Assurance Activity Report

Fortinet, Inc.
FortiSandbox 4.4

VID 11636

UL15437480-AAR V1.3
October 30, 2025

Evaluated by:

UL Verification Services Inc.
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

TOE Evaluation Sponsor and Developer:
Fortinet, Inc.
899 Kifer Rd
Sunnyvale, CA 94086

ST Author:
Brad Mitchell
UL Verification Services Inc.
709 Fiero Lane, Suite 25
San Luis Obispo, CA 93401

Evaluation Personnel:
Michael Baron
Meghana Achanta
Dylan Lyman
Sr. Review:
Oleg Andrianov

Applicable Common Criteria Version
CC Version 3.1 R5, April 2017

Common Evaluation Methodology Version
CEM Version 3.1 R5, April 2017

Applicable Common Criteria Protection Profiles
collaborative Protection Profile for Network Devices
Version 3.0e, December 06, 2023

Table of Contents

1 Overview

This document presents evaluation results of the FortiSandbox 4.4 against the collaborative Protection Profile for Network Devices, Version 3.0e, December 06, 2023, and Functional Package for Secure Shell version 1.0, 2021]. This document contains a description of the assurance activities and associated results as performed by UL, an accredited Common Criteria Testing Laboratory. This Evaluation was conducted with the oversight and guidance provided by the National Information Assurance Partnership and its contributors.

1.1 Tested platforms

The [ST] claims three (3) physical models and one (1) virtual model:

- FortiSandbox 1500G, using an AMD EPYC 7313P CPU on the Zen 3 Microarchitecture
- FortiSandbox 500G, using an AMD EPYC 3251 CPU on the Zen microarchitecture
- FortiSandbox 3000F, using an AMD EPYC 7402 CPU on the Zen 2 microarchitecture
- FortiSandbox 4.4.6-build 4510 VM image for ESXi virtualization servers

The evaluator performed testing on the FSA-1500G model physical network device and on the virtual network device running the same firmware.

1.1.1 Rationale Regarding Testing equivalence:

The differences between the hardware platforms are as follows:

TOE Model	CPU	Network Interfaces	Storage	Physical Format
FortiSandbox FSA-500G	AMD EYPC 3251 CPU on the Zen microarchitecture	4x GE RJ45 ports 1 RJ45 Console Port	1x 960 GB	1RU
FortiSandbox 1500G	AMD EYPC 7313P CPU on the Zen 3 microarchitecture	4x GE RJ45 ports, 2x 10 GE SFP+ slots 1 RJ45 Console Port	2x 960 GB RAID1	1RU
FortiSandbox FSA-3000F	AMD EYPC 7402 CPU on the Zen 2 microarchitecture	4x GE RJ45 ports, 2x 10 GE SFP+ slots 1 RJ45 Console Port	4x 2 TB RAID-10	2RU

Security relevance of differences rationale:

- CPU
  - Each model uses a different CPU as listed above. These differences are significant to cryptographic tests only. For full test coverage, CAVP testing was conducted on each model in the TOE.
- Physical size
  - No security relevant difference
- Data Storage Size
  - Not security relevant since the only TSF that is dependent on the amount of storage is FAU_STG and the minimum amount of storage provided out of the TOE models is sufficient to cover this TSF.
- Network Interfaces
  - Number of interfaces; no security relevance.
  - Type of interfaces; no security relevance as OSI layer 1 and layer 2 differences do not impact layer 3 and above functionality.

1.2 Test Environment overview

The test environment used by the CCTL during the course of testing is briefly summarized below and conforms to the expected use-case of the TOE as Physical Network Device, Virtual Network Device.

The evaluation team performed the independent testing activities to confirm the TOE operates to the TOE security functional requirements as specified in the [ST] for a product claiming conformance to [PP] and [PKG]. The evaluation team devised a Test Plan based on the Testing Assurance Activities specified in [PP] and [PKG]. The Test Plan described how each test activity was to be performed. The evaluation team executed the tests specified in the Test Plan and documented the results in the Evaluation Technical Report. The evaluation team consisted of ***Michael C. Baron, Dylan Lyman, Meghana Venkata Achanta***, senior review by ***Oleg Andrianov***, from the CCTL.
The test laboratory was configured by UL and is physically located at the UL San Luis Obispo facility in an access-controlled environment.

1.3 Test environment for pND model.

The CC test bed consists physically of:

Number	Device	Purpose
1	HP ProLiant Blade Server	ESXi 8.0.3 Server w/8 LAN ports
1	Aruba 2920-24G 24-port layer 3 aware Ethernet switch	Infrastructure Routing Switch
1	Dell PowerEdge Desktop Workstation	Management & Control console

Overview

The test lab can present three separate test environments (referred to as strings) via the extensive use of virtualization. Access to these environments is through the management console.
The management console provides the access point for the evaluator either through physical access in the lab or via SSH or remote desktop. These provide the ability to subsequently connect to the various devices in each string without exposing the entire environment to the external network. Each workstation has one network interface on the IGL test network and a second interface connected to the Aruba 2920 switch. Each testing string, depending on TOE type, may contain up to 4 subnets:

- Management: Network segment used for remote (either public or private facing subnet) administration of the TOE and the VMs in the environment.
- Private: Network segment notionally attached to the LAN side of the TOE.

The HP ProLiant blade server provides virtual machines for each of the test strings, which may contain the following virtual hosts:

- Test Host – Used in testing TLS and SSH protocols.
- FortiAnalyzer – Used to capture remote audit logs.

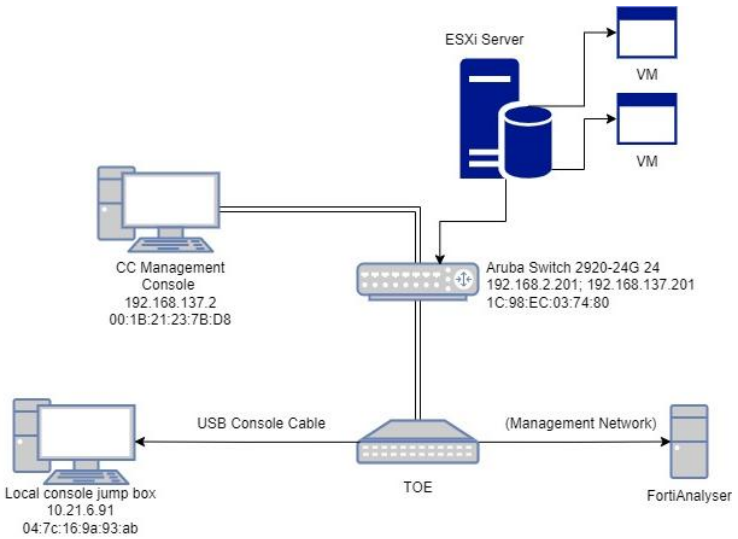

Figure 1 – pND Test Network Configuration

Two test strings were utilized to test two of the same TOE models so that multiple evaluators could test simultaneously. The test strings used for this evaluation consist of the following virtual and physical networks:

String 3:

Network name	Description	Addressing scheme
Management	TOE Trusted Path & Trusted Channel	192.168.3.0/24

String 4:

Network name	Description	Addressing scheme
Management	TOE Trusted Path & Trusted Channel	192.168.4.0/24

The detailed test report contains detailed descriptions of the test network and the test machines in it, as required by Labgram #111.

1.4 Test environment for vND model.

The CC test bed consists physically of:

Number	Device	Purpose
1	HP ProLiant Blade Server	ESXi 8.0.3 Server w/8 LAN ports
1	Aruba 2920-24G 24-port layer 3 aware Ethernet switch	Infrastructure Routing Switch
1	Dell PowerEdge Desktop Workstation	Management & Control console

Figure 2 - ESXi Version information

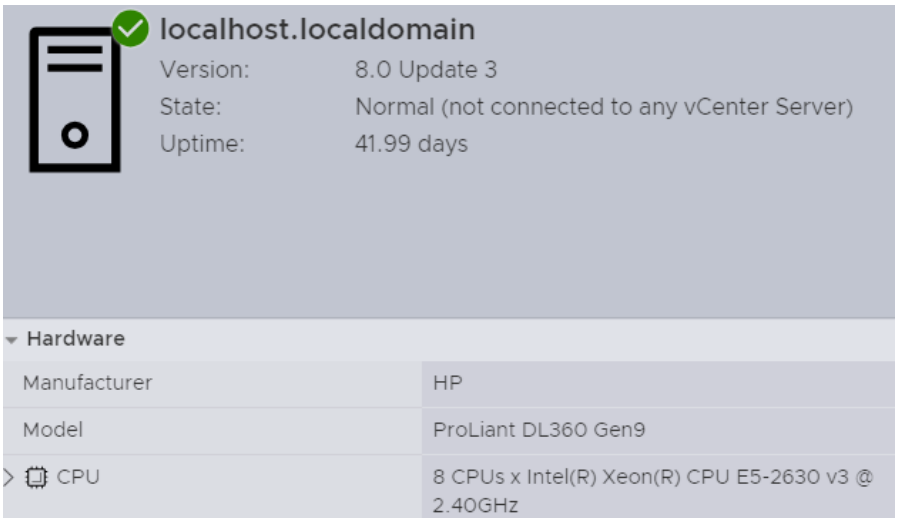

Figure 3 - ESXi Hardware information

Overview

The test lab can present three separate test environments (referred to as strings) via the extensive use of virtualization. Access to these environments is through the management console.

The management console provides the access point for the evaluator either through physical access in the lab or via SSH or remote desktop. These provide the ability to subsequently connect to the various devices in each string without exposing the entire environment to the external network. Each workstation has one network interface on the test network and a second interface connected to the Aruba 2920 switch. Each testing string, depending on TOE type, may contain up to 4 subnets:

- Management: Network segment used for remote (either public or private facing subnet) administration of the TOE and the VMs in the environment.
- Private: Network segment notionally attached to the LAN side of the TOE.

The HP ProLiant blade server provides virtual machines for each of the test strings, which may contain the following virtual hosts:

- Test Host – Used in testing the TLS and SSH protocols.
- FortiAnalyzer – Used to capture remote audit logs.

Figure 4 – vND Test Network Configuration shows the logical configuration used for the ATE.

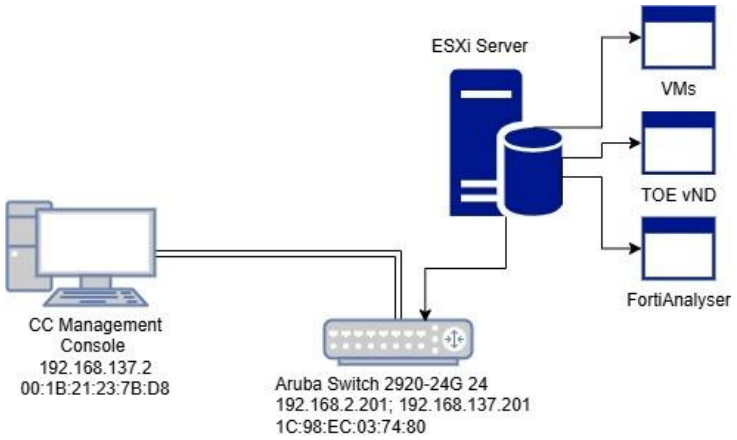

Figure 4 – vND Test Network Configuration

The test report contains detailed description of the test network, and the test machines in it, as required by Labgram #111.

1.5 Delivery and physical setup

The TOE was obtained from the courier delivery method as described in [AGD] Section “Verifying secure delivery”.

Setup was performed in accordance with [AGD] instructions as identified in the steps for ATE_IND.1-2 found in this report.

Guidance was obtained from the vendor directly for the purposes of this evaluation. The final guidance documentation will be posted on the vendor’s website and available for download by customers, post successful evaluation.

The TOE’s firmware was obtained from the vendor directly for the purposes of this evaluation. The final firmware build will be posted on the vendor’s website and available for download by customers, post successful evaluation.

1.6 Initial Configuration

The evaluator followed steps in [AGD] Section “Installing the unit” to perform the initial installation of the TOE.

2 SFR Assurance Activities and Results

2.1 FAU_GEN.1 Audit Data Generation

TSS

For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS shall identify what information is logged to identify the relevant key.

For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

[ST] Section 7.1.1 identifies what information is logged to identify the relevant key. Associated creation details (date, time, user) for SSH, key generation is logged with identifying information, admin user and IP address. CSR-related keypairs are identified by generation date and time, and user who generated the CSR.

Guidance Documentation

The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

An example audit record is included in guidance document, “NDcPP Common Criteria Logging Addendum,” for each required auditable event.

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

The evaluators ensured through review of the log example documentation that an audit event record was included for each of the required auditable events.

Tests

The evaluator shall test the TOE’s ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different identity and authentication (I&A) mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

2.1.1 Tests

Test Number	1
Test Objective	Verify that the TOE generates audit records for the required events.
Test Steps Performed	The evaluator exercised the TOE either independently or as part of evaluation testing to generate audit records for all required audit events. These records were found to meet the requirement. The TOE is not a distributed TOE.
Test Result	Pass

2.2 FAU_GEN.2 User Identity Association

TSS & Guidance Documentation

The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

The TOE is not a distributed TOE.

Tests

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.

The TOE is not a distributed TOE.

2.3 ~This section left blank intentionally~

2.4 FAU_STG.1 Protected Audit Trail Storage (Optional)

TSS

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how local storage is implemented among the different TOE components (e.g. every TOE component does its own local storage or the data is sent to another TOE component for central local storage of all audit events).

[ST] Section 7.1.2 describes the amount of audit data that are stored locally, how these records are protected against unauthorized modification or deletion, and the conditions that must be met for authorized deletion of audit records.

Guidance Documentation

The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

[AGD] does not state that configuration is necessary for the secure storage of locally stored audit data.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall attempt to access the audit trail without authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Test 2: The evaluator shall access the audit trail as an authorized administrator and attempt to delete the audit records (if supported by the TOE, and to the extent described in the TSS). The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.

2.4.1 Tests

Test Number	1
Test Objective	Verify that no user can access locally stored audit records without prior authentication as a security administrator. Verify that authenticated, non-security administrator users are unable to access the locally stored audit records.
Test Steps Performed	The evaluator created and authenticated as a user without log access privileges and found that log records were inaccessible.
Test Result	Pass

Test Number	2
Test Objective	Verify an authenticated user with administrative privileges is able to delete locally stored audit records.
Test Steps Performed	The evaluator authenticated as an admin user with log management permissions and was able to delete locally stored audit records.
Test Result	Pass

2.5 FAU_STG_EXT.1 Protected Audit Event Storage

TSS

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

[ST] Section 7.1.2 describes the means by which the audit data are transferred to the external audit server. [ST] Section 7.7 describes how the trusted channel is provided.

The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

[ST] Section 7.1.1 describes that the TOE is a standalone TOE that stores audit data locally.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real- time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator shall verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

[ST] Section 7.1.2 describes that the transmission of audit data to an external IT entity can be done in real time as long as the trusted channel is established. The TOE is not capable of performing transmission periodically.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

The TOE is not distributed.

The evaluator shall examine the TSS to ensure it describes the amount of audit data that can be stored locally and how these records are protected against unauthorized modification or deletion.

This is covered by FAU_STG.1 TSS evaluation activities.

The evaluator shall examine the TSS to ensure it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent.

[ST] Section 7.1.2 describes local logging as being stored in a file system as a persistent database file.

The evaluator shall examine the TSS to ensure it describes the conditions that must be met for authorized deletion of audit records.

This is covered by FAU_STG.1 TSS evaluation activities.

The evaluator shall examine the TSS to ensure it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

[ST] Section 7.1.2 describes, “The TOE maintains 10 log files of 100mb each. When a log file reaches 100mb, it is “closed” and a new log file is opened. When the TOE needs to close the 10th log file, the oldest (“first”) log file is deleted, and a new file is opened.”

For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

The TOE is not a distributed TOE.

Guidance Documentation

The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

The evaluator shall also examine the guidance documentation to ensure it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and “cleared” periodically by sending the data to the audit server.

The evaluator shall examine the guidance documentation to ensure it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

If the storage size is configurable, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying the required parameters.

If more than one selection is made for FAU_STG_EXT.1.5, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying which action is performed when the local storage space is full.

[AGD] Section “Log Specific Settings” and [FSAG] Section “Log Servers” contain the instructions to establish the trusted channel from the TOE to an audit server. [FSAG] Section “Log Servers” outlines the requirements for a log server. [AGD] Section “FortiAnalyzer configuration” describes how to configure and establish a connection with the Fortinet proprietary log server, FortiAnalyzer.
[AGD] Section “Local Logging” describes the storage of local audit data.
[AGD] does not state that configuration is necessary for the secure storage of locally stored audit data.

Tests

Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s).

The evaluator shall perform the following additional test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator’s choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

Test 2: For distributed TOEs, Test 1 defined above shall be applicable to all TOE components that forward audit data to an external audit server.

Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions:

If persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.

If non-persistent logging is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are no longer present within the local audit storage.

Test 4: The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that:

The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.5).

The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.5)

The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.5).

Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2, Test 2 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

Test 6 [Conditional]: In case manual export or ability to view locally is selected in FAU_STG_EXT.1.6, during interruption the evaluator shall perform a TSF-mediated action and verify the event is recorded in the audit trail.

2.5.1 Tests

Test Number	1
Test Objective	Verify the audit data from the TOE is being sent as encrypted data. Verify audit logs are sent automatically and without administrator intervention.
Test Steps Performed	The evaluator conducted a packet capture on the log server packets, which revealed encrypted packets.
Test Result	Pass

Test Number	2
Test Objective	N/A
Test Steps Performed	The TOE is not distributed.
Test Result	Pass

Test Number	3
Test Objective	Verify that local audit records persist after reboot of the TOE.
Test Steps Performed	The evaluator rebooted the TOE and verified that audit records recorded in previous testing were still present.
Test Result	Pass

Test Number	4
Test Objective	Verify the TOE overwrites the oldest log data once the local audit storage is filled to capacity.
Test Steps Performed	The evaluator made a note of the oldest log data and executed a script to overload the log buffer. It was then observed that the oldest log data was overwritten once capacity had been reached.
Test Result	Pass

Test Number	5
Test Objective	N/A
Test Steps Performed	The TOE is not distributed.
Test Result	Pass

Test Number	6
Test Objective	Both manual export and ability to view locally stored audit records are selected in [ST]. Logically disconnect the transmission of audit records to the remote audit server, perform action to generate audit record on the TOE, and verify that the audit record is stored locally via the Web GUI interface.
Test Steps Performed	The evaluator logically disconnected the TOE's connection to the audit server, then generated an audit record and observed that the audit record was stored locally without transmission.
Test Result	Pass

2.6 FCS_CKM.1 Cryptographic Key Generation

TSS

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

[ST] Section 7.2.1 identifies the key sizes as 2048 bits for RSA schemes, P-251, P-384, P521 for ECC schemes used in TLS client/server SSH and X.509 certificate requests, 2048 bits for FFC schemes used in TLS client and FCC schemes using safe-prime groups used in TLS server.

Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

[AGD] Section "The CC Mode of Operation" describes how to configure the TOE to use selected key generation schemes and key sizes by enabling the CC mode of operation. The [AGD] notes "…restrictive default settings are implemented." Enabling CC mode restricts to [ST] selects key generation schemes. [AGD] Section "Administration" outlines the restricted ciphersuites enabled by CC mode of operation.

Tests

Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported).

Key Generation for FIPS PUB 186-4 RSA Schemes

The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include:

Random Primes:

Provable primes

Probable primes

Primes with Conditions:

Primes p1, p2, q1, q2, p and q shall all be provable primes

Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes

Primes p1, p2, q1, q2, p and q shall all be probable primes

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.

The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

Primes q and p shall both be provable primes

Primes q and field prime p shall both be probable primes

and two ways to generate the cryptographic group generator g:

Generator g constructed through a verifiable process

Generator g constructed through an unverifiable process.

The Key generation specifies 2 ways to generate the private key x:

len(q) bit output of RBG where 1 <=x <= q-1

len(q) + 64 bit output of RBG, followed by a mod q-1 operation and a +1 operation, where 1<= x<=q-1.

The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

g != 0,1

q divides p-1

g^q mod p = 1

g^x mod p = y

for each FFC parameter set and key pair.

FFC Schemes using "safe-prime" groups

Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

2.6.1 Tests

Test Number	1
Test Objective	Verify that the TOE correctly implements the following crypto schemes: • RSA schemes using cryptographic key sizes of 2048 bits that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3; • ECC schemes using 'NIST curves' P-256, P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4; • FFC schemes using cryptographic key sizes of 2048 bits that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1; • FFC Schemes using 'safe-prime' groups (2048 bits) that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 3526.
Test Steps Performed	CAVP Certificates A7082, A7080 cover key generation for RSA, ECC, FFC and FFC Schemes using 'safe-prime' groups.
Test Result	Pass.

2.7 FCS_CKM.2 Cryptographic Key Establishment

TSS

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. It is sufficient to provide the scheme, SFR, and service in the TSS.

The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be as shown in the table below. The information provided in the example above does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_IPSEC_EXT.1	Authentication Server

[ST] Section 7.2.1 describes the following cryptographic key establishment schemes and associates them with the following TSF:

Scheme	SFR	Service
ECDSA (P-256, P-384, P-521)	FCS_TLSS_EXT.1;	Administration
FFC Schemes using ‘safe-prime’ groups (2048-bit MODP)	FCS_TLSS_EXT.1; FCS_SSH_EXT.1; FCS_SSHS_EXT.1	Administration
ECDSA (P-256, P-384, P-521)	FCS_TLSC_EXT.1;	Audit Server

Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

[AGD] Section “The CC Mode of Operation” describes how to configure the TOE to use selected key generation schemes and key sizes by enabling the CC mode of operation. The [AGD] notes “…restrictive default settings are implemented.” Enabling CC mode restricts to [ST] selected key establishment schemes.
[AGD] Section “Administration” outlines the restricted ciphersuites enabled by CC mode of operation.

Tests

Key Establishment Schemes

The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below.

*ECC and FIPS 186-type FFC **SP800-56A Key Establishment Schemes***

The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been

implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

RSA-based key establishment

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

FFC Schemes using "safe-prime" groups

The evaluator shall verify the correctness of the TSF's implementation of safe- prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

2.7.1 Tests

Test Number	1
Test Objective	Verify key establishment schemes claimed by the TOE, using the following cryptographic Elliptic curve-based scheme ECDSA).
Test Steps Performed	CAVP Certificates A7080 and A7082 cover key agreement for Elliptic curve-based scheme (KAS ECC) P-256, P-384, P-521, and FFC schemes using safe prime groups (MODP 2048).
Test Result	Pass

2.8 FCS_CKM.4 Cryptographic Key Destruction

TSS

The evaluator shall examine the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator shall confirm that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for[1]). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.

The Table 6 in [ST] Section 7.2.1, lists all relevant keys and identifies the required details. The evaluator confirmed that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols).

[ST] Section 7.2.1, Table 7 includes details for "User/Admin Passwords" for the FPT_APW_EXT.1 TSF.

The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory, the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

[1] Where keys are stored encrypted or wrapped under another key then this may need to be explained in order to allow the evaluator to confirm the consistency of the description of keys with the TOE functions.

[ST] Section 7.2.1 states “Each of the CSPs are protected from unauthorized access via memory management which disallows any memory reads from other processes within the OS ensuring that the CSPs are only available to the calling application”, and “Plaintext private keys for the purposes of SSH, HTTPS and TLS are maintained on the flash filesystem and are not viewable through the TOE interfaces”, and finally “The TOE does not provide any interfaces to view the keys/CSPs” - which all address the FPT_SKP_EXT.1 TSF.

[ST] Section 7.2.1 identifies how the TOE destroys keys stored as plaintext in non-volatile memory to include the following required detail:

- identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Where the ST specifies the use of “a value that does not contain any CSP” to overwrite keys, the evaluator shall examine the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

The TSS does not identify configurations or circumstances that may not conform to the key destruction requirement.

Guidance Documentation

A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command[2] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

[AGD] Section “Key Zeroization” describes the process that the TOE utilizes for key destruction (i.e., All keys and CSPs are zeroized by initializing a factory reset.
The [ST] TSS is consistent with the [AGD] description.

2.8.1 Tests

No test activity

2.9 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

TSS

[2] Where TRIM is used then the TSS and/or guidance documentation is also expected to describe how the keys are stored such that they are not inaccessible to TRIM, (e.g. they would need not to be contained in a file less than 982 bytes which would be completely contained in the master file table).

The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

[ST] Section 7.2.1, Table 6 identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption, to include the following details:

- AES in CBC mode (128, 256 bits)
- AES in GCM mode (128, 256 bits)

Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

[AGD] Section "The CC Mode of Operation" describes how to configure the TOE to use selected modes and key sizes for data encryption/decryption by enabling the CC mode of operation. The [AGD] notes "…restrictive default settings are implemented." Enabling CC mode restricts to [ST] selected modes and key sizes.

Tests

AES-CBC Known Answer Tests

There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES- CBC decryption.

KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of keys and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit

key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of keys and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for I in [1,N].The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.

KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256- bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

 # Input: PT, IV, Key
for i = 1 to 1000:

if i == 1:

CT[1] = AES-CBC-Encrypt(Key, IV, PT)

PT = IV

else:

CT[i] = AES-CBC-Encrypt(Key, PT)

PT = CT[i-1]

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES- CBC-Decrypt.

AES-GCM Test

The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

128 bit and 256 bit keys

Two plaintext lengths. One of the plaintext lengths shall be a non- zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-CTR Known Answer Tests

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter if the TSF is validated against the requirements of the Functional Package for Secure Shell referenced in Section 2.2 of the cPP. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES- GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm):

There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros.

KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128].

AES-CTR Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

 # Input: PT, Key
for i = 1 to 1000:
CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i]

The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall perform this test using each selected keysize.

There is no need to test the decryption engine.

2.9.1 Tests

Test Number	1
Test Objective	Verify that the TOE correctly implements data encryption according to AES in CBC or GCM modes.
Test Steps Performed	CAVP Certificates A7082, A7080, A7083 and A7081 cover data encryption algorithms used by the TOE.
Test Result	Pass

2.10 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

TSS

The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

[ST] Section 7.2.1, Table 6 specifies the cryptographic algorithm and key size supported by the TOE for signature services, to include the following details:

- RSA
    - 2048-bit modulus

- ECDSA
    - P-256, P-384, P-521

[ST] Section 7.2.2 states “RSA signature generation and verification (2048-bit)”

Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

[AGD] Section “The CC Mode of Operation” describes how to configure the TOE to use selected signature schemes and key sizes by enabling the CC mode of operation. The [AGD] notes “…restrictive default settings are implemented.” Enabling CC mode restricts to [ST] selected cryptographic algorithms and key sizes.

Tests

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.

ECDSA FIPS 186-4 Signature Verification Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

RSA Signature Algorithm Tests

Signature Generation Test

The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures.

The evaluator shall verify the correctness of the TOE’s signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures.

Signature Verification Test

For each modulus size/hash algorithm selected, the evaluator shall generate a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results.

The evaluator shall verify that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

2.10.1 Tests

Test Number	1
Test Objective	Verify that the TOE correctly implements crypto function RSA and ECDSA Schemes.
Test Steps Performed	CAVP Certificates A7080 and A7082 cover key generation for RSA (2048-bit) and ECDSA (P-256, P-384, P-521).
Test Result	Pass

2.11 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

TSS

The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

[ST] Section 7.2.2 describes the association of the hash function with other TSF cryptographic functions, with the following details:

Hash Algorithm	TSF Usage
SHA-1	In TLS ciphersuites In matching HMAC
SHA2-512	SSH session message integrity verification Creating hashes of user passwords for storage in the underlaying file system. In matching HMAC In TLS signature algorithms
SHA2-384	In TLS ciphersuites In TLS Signature algorithms In matching HMAC
SHA2-256	SSH session message integrity verification SSH session key establishment SSH peer authentication when the TOE is a server In TLS Ciphersuites In TLS signature algorithms Self-test of configuration integrity Firmware update integrity verification In matching HMAC

Guidance Documentation

The evaluator shall check the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

[AGD] Section “The CC Mode of Operation” describes how to configure the TOE to use [ST] selected hash sizes by enabling the CC mode of operation. The [AGD] notes “…restrictive default settings are implemented.” Enabling CC mode restricts to [ST] selected hash sizes.

Tests

The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bitoriented vs. the byteoriented testmacs.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

Short Messages Test - Bit-oriented Mode

The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluator shall compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test - Byte-oriented Mode

The evaluator shall devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Bit-oriented Mode

*The evaluator shall devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

Selected Long Messages Test - Byte-oriented Mode

*The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*

Pseudorandomly Generated Messages Test

This test is for byteoriented implementations only. The evaluator shall randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

2.11.1 Tests

Test Number	1
Test Objective	Verify that the TOE correctly implements crypto function SHA and SHA-2 Schemes.
Test Steps Performed	CAVP Certificates A7082, A7080, A7083 and A7081 cover SHA-1, SHA-256, SHA-384, SHA-512.
Test Result	Pass

2.12 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

TSS

The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

[ST] Section 7.2.2 specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used, with the following details:

- HMAC-SHA-1 (160 bit key size with 512 bit block size, 160 bit output length),
- HMAC-SHA-256 (256 bit keysize with 512 bit block size, 256 bit output length),
- HMAC-SHA-384 (256 bit key with a 1024 bit block size, 384 bit output length)
- HMAC-SHA-512 (256 bit key with 1024 bit block, 512 bit output length)

Guidance Documentation

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

[AGD] Section "The CC Mode of Operation" describes how to configure the TOE to use selected key hash functions by enabling the CC mode of operation. The [AGD] notes "…restrictive default settings are implemented." Enabling CC mode restricts to [ST] selected key hash functions.

Tests

For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.

2.12.1 Tests

Test Number	1
Test Objective	Verify that the TOE correctly implements crypto function HMACs.
Test Steps Performed	CAVP Certificates A7082, A7080, A7083 and A7081 covers HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512.
Test Result	Pass

2.13 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix D of [NDcPP].

TSS

The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min- entropy contained in the combined seed value.

[ST] Section 7.2.1 specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value, with the following details:

• CTR_DRBG (AES-256)

• JitterEntropy 3.4.1 as a raw entropy source collected from CPU execution time jitter

• Noise source provides full entropy to seed the DRBG with 256 bits

Guidance Documentation

The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

[AGD] “RBG Seeding and Reseed Interval” describes the CLI command to configure the reseed period of the RBG.

Tests

The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator shall verifiy that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator shall verifiy that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

***Entropy input:** the length of the entropy input value must equal the seed length.*

***Nonce:** If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.*

***Personalization string:** The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.*

***Additional input:** the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.*

2.13.1 Tests

Test Number	1
Test Objective	Verify the TOE RNG is working correctly. This test is resolved by a CAVP certificate.
Test Steps Performed	TSF utilizes a counter-DRBG utilizing AES-256. This claim is supported by certificate A7083 and A7081.
Test Result	Pass

2.14 FCS_HTTPS_EXT.1 HTTPS Protocol (Selection-Based)

TSS

The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

[ST] Section 7.2.5 states “The TOE’s HTTPS protocol complies with RFC 2818 (by implementing all “SHALL” and “MUST” statements) and uses the TLS functionality described below.)”

Guidance Documentation

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

[AGD] Section “Remote access requirements” describes how, by enabling the CC mode of operation, remote administration is restricted to HTTPS by default.

Tests

This test is now performed as part of FIA_X509_EXT.1/Rev testing.

Tests are performed in conjunction with the TLS evaluation activities.

If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

2.14.1 Tests

Test Number	1
Test Objective	This test is satisfied by the tests performed for FIA_X509_EXT.1/Rev. The TSF includes an HTTPS server, but it does not support client authentication for it.
Test Steps Performed	N/A
Test Result	Pass

2.15 FCS_SSH_EXT.1 SSH Protocol

2.15.1 FCS_SSH_EXT.1.1

TSS

The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this and subsequent components. Otherwise, this SFR is evaluated by activities for other SFRs.

[ST] Section 6.2.10 contains the selections for this SFR, with the following details:

- SSH as server
- RFCs:
  - 4251 (mandatory)
  - 4252 (mandatory)
  - 4253 (mandatory)
  - 4254 (mandatory)
  - 4256 (missing) - keyboard-interactive authentication
  - 6668 - HMAC-SHA-2 algorithms

- 8268 - FFC DH groups with SHA-2
- 8308 - if RFC 8332 is selected
- 8332 - SHA-2 is available with ssh-rsa

The selections in this SFR are consistent with subsequent components of the SSH SFRs.

Guidance Documentation

There are no guidance evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

Tests

There are no test evaluation activities for this component. This SFR is evaluated by activities for other SFRs.

2.15.2 FCS_SSH_EXT.1.2

TSS

The evaluator shall check to ensure that the authentication methods listed in the TSS are identical to those listed in this SFR component; and, ensure if password-based authentication methods have been selected in the ST then these are also described; and, ensure that if keyboard-interactive is selected, it describes the multifactor authentication mechanisms provided by the TOE.

[ST] Section 6.2.10 contains the selection for this SFR - “keyboard-interactive” (RFC 4256). [ST] Section 7.2.3 describes the supported authentication methods – “The TOE supports SSH password-based authentication method as described in RFC 4256”. These sections are consistent with each other.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure the configuration options, if any, for authentication mechanisms provided by the TOE are described.

[AGD] Section “Enabling administrative access via SSH” describes the process in which to enable SSH while in the CC mode of operation. The CC mode of operation ensures the TOE restricts configuration to [ST] selections.

Tests

Test 1[conditional]: If the TOE is acting as SSH Server:

The evaluator shall use a suitable SSH Client to connect to the TOE, enable debug messages in the SSH Client, and examine the debug messages to determine that only the configured authentication methods for the TOE were offered by the server.

[conditional] If the SSH server supports X509 based Client authentication options:

The evaluator shall initiate an SSH session from a client where the username is associated with the X509 certificate. The evaluator shall verify the session is successfully established.

Next the evaluator shall use the same X509 certificate as above but include a username not associated with the certificate. The evaluator shall verify that the session does not establish.

Finally, the evaluator shall use the correct username (from step a above) but use a different X509 certificate which is not associated with the username. The evaluator shall verify that the session does not establish.

Test 2 [conditional]: If the TOE is acting as SSH Client, the evaluator shall test for a successful configuration setting of each authentication method as follows:

The evaluator shall initiate a SSH session using the authentication method configured and verify that the session is successfully established.

Next, the evaluator shall use bad authentication data (e.g. incorrectly generated certificate or incorrect password) and ensure that the connection is rejected.

Steps a-b shall be repeated for each independently configurable authentication method supported by the server.

Test 3 [conditional] If the TOE is acting as SSH Client, the evaluator shall verify that the connection fails upon configuration mismatch as follows:

The evaluator shall configure the Client with an authentication method not supported by the Server.

The evaluator shall verify that the connection fails.

If the Client supports only one authentication method, the evaluator can test this failure of connection by configuring the Server with an authentication method not supported by the Client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.

Test Number	1
Test Objective	Verify that the TOE supports only the keyboard interactive (RFC 4256)" user authentication method. Note: The conditional test 'b.' is not applicable to the TOE.
Test Steps Performed	The evaluator configured an SSH client to enable debug messages and connected to the TOE. The evaluator verified that the TOE offered only the keyboard-interactive authentication mechanisms as claimed in the ST.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE supports configuration of authentication methods.
Test Steps Performed	This test is not applicable, as the TOE does not claim SSH as a client.
Test Result	Pass

Test Number	3
Test Objective	This test is not applicable, as the TOE does not claim SSH as a client.
Test Steps Performed	N/A
Test Result	Pass

2.15.3 FCS_SSH_EXT.1.3

TSS

The evaluator shall check that the TSS describes how "large packets" are detected and handled.

[ST] Section 7.2.3 describes how "large packets" are detected and handled, with the following details:

- "At all times, packets greater than 256K bytes in an SSH transport connection are dropped by terminating the SSH session and not processing the oversize packet."

Guidance Documentation

None.

Tests

Test 1: The evaluator shall demonstrate that the TOE accepts the maximum allowed packet size. *Test 2: This test is performed to verify that the TOE drops packets that are larger than size specified in the component.* *The evaluator shall establish a successful SSH connection with the peer.* *Next the evaluator shall craft a packet that is slightly larger than the maximum size specified in this component and send it through the established SSH connection to the TOE. The packet should not be greater than the maximum packet size + 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet.* *The evaluator shall verify that the packet was dropped by the TOE. The method of verification will vary by the TOE. Examples include reviewing the TOE audit log for a dropped packet audit or observing the TOE terminates the connection.*

Test Number	1
Test Objective	Verify that the TOE's SSH server implementation accepts its maximum SSH packet of size 256,000 bytes.
Test Steps Performed	The evaluator used a proprietary testing tool to send a packet of the maximum size permitted by the TOE. The evaluator verified that the TOE accepted this packet.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE either silently drops SSH packets larger than 262,144 bytes or terminates the SSH connection when receiving an SSH packet larger than 262,144 bytes.
Test Steps Performed	The evaluator utilized a proprietary testing tool to send a packet just larger than the maximum packet size. The evaluator observed that the TOE terminated the connection.
Test Result	Pass

2.15.4 FCS_SSH_EXT.1.4

TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

[ST] Section 7.2.3 describes the encryption algorithms supported, with the following details:

- "the TOE uses aes128-CBC and aes256-CBC data encryption modes, rejecting all other encryption algorithms"

The description is consistent with the selections for this SFR.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

[AGD] Section “Enabling administrative access via SSH” describes the process in which to enable SSH while in the CC mode of operation. The CC mode of operation ensures the TOE restricts configuration to [ST] selections.

Tests

The evaluator shall perform the following tests.

If the TOE can be both a client and a server, these tests must be performed for both roles.

Test 1: The evaluator must ensure that only claimed algorithms and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall establish an SSH connection with a remote endpoint. The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers only the algorithms defined in the ST for the TOE for SSH connections. The evaluator shall perform one successful negotiation of an SSH connection and verify that the negotiated algorithms were included in the advertised set. If the evaluator detects that not all algorithms defined in the ST for SSH are advertised by the TOE or the TOE advertises additional algorithms not defined in the ST for SSH, the test shall be regarded as failed.

The data collected from the connection above shall be used for verification of the advertised hashing and shared secret establishment algorithms in FCS_SSH_EXT.1.5 and FCS_SSH_EXT.1.6 respectively.

Test 2: For the connection established in Test 1, the evaluator shall terminate the connection and observe that the TOE terminates the connection.

Test 3: The evaluator shall configure the remote endpoint to only allow a mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.

Test Number	1
Test Objective	By capturing the network traffic between the TOE and ENV3_SSH_Test_Host VM while successfully establishing an SSH session, verify that only the algorithms and cryptographic primitives claimed in [ST] are used to establish an SSH session.
Test Steps Performed	The evaluator established an SSH connection while capturing packets. The evaluator inspected the packet capture and verified that only the aes128-cbc and aes256-cbc encryption algorithms were included in the advertisement, as specified in the ST.
Test Result	Pass

Test Number	2
Test Objective	The evaluator verifies that the Security Administrator can terminate their own session by typing “exit” at the successfully authenticated SSH CLI session.
Test Steps Performed	The evaluator issued the ‘exit’ command on the connection established in Test 1 and verified that the session was terminated.
Test Result	Pass

Test Number	3
Test Objective	The evaluator verifies that the TOE rejects an SSH session establishment attempt when using the TSF-unsupported cipher of aes128-ctr.

Test Steps Performed	The evaluator configured an SSH client to allow only the non-selected aes128-ctr cipher. The evaluator attempted to connect to the TOE and verified that the connection failed.
Test Result	Pass

2.15.5 FCS_SSH_EXT.1.5

TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator will check the TSS to ensure that the hashing algorithms specified are identical to those listed for this component.

[ST] Section 6.2.10 contains the selections for this SFR:
- hmac-sha2-256 (RFC 6668)
- hmac-sha2-512 (RFC 6668)

[ST] Section 7.2.3 describes the keyed-hashing algorithms supported, with the following details:
- "The SSH transport implementation uses hmac-sha2-256 and hmac-sha2-384 as its data integrity MAC algorithms and rejects all other MAC algorithms."

These sections are consistent with each other.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

FCS_SSH_EXT.1.2, FCS_SSH_EXT.1.4, FCS_SSH_EXT.1.6

[AGD] Section "Enabling administrative access via SSH" describes the process in which to enable SSH while in the CC mode of operation. The CC mode of operation ensures the TOE restricts configuration to [ST] selections.

FCS_SSH_EXT.1.8

[AGD] Section "SSH Rekey" describes the TSF rekey behavior for SSH.

Tests

Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

Test 2: The evaluator shall configure an SSH peer to allow only a hashing algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

Test Number	1
Test Objective	Verify that the TOE proposed only hmac-sha2-256 and hmac-sha2-512 MAC algorithms.
Test Steps Performed	The evaluator inspected the test data collected in FCS_SSH_EXT.1.4 Test 1 and verified that only the hmac-sha2-256 and hmac-sha2-512 MAC algorithms were present, as claimed in the ST.
Test Result	Pass

Test Number	2
Test Objective	The evaluator verifies that the TOE rejects an SSH session establishment attempt when using the TSF-unsupported HMAC algorithm of hmac-sha1-96.
Test Steps Performed	The evaluator configured an SSH peer to allow only the non-claimed hmac-sha1 MAC algorithm. The evaluator verified that the connection failed.
Test Result	Pass

2.15.6 FCS_SSH_EXT.1.6

TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified. The evaluator will check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this component.

[ST] Section 6.2.10 contains the selections for this SFR:

- diffie-hellman-group14-sha256 (RFC 8268)

[ST] Section 7.2.3 describes the shared secret establishment algorithms supported, with the following details:

- "The TOE's SSH implementation uses only diffie-hellman-group14-sha256_for its key exchange methods."

These sections are consistent with each other.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

[AGD] Section "Enabling administrative access via SSH" describes the process in which to enable SSH while in the CC mode of operation. The CC mode of operation ensures the TOE restricts configuration to [ST] selections.

Tests

Test 1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

Test 2: The evaluator shall configure an SSH peer to allow only a key exchange method that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.

Test Number	1
Test Objective	Verify that the TOE proposes only the diffie-hellman-group14-sha256 key exchange algorithm.
Test Steps Performed	The evaluator inspected the data captured in the evidence for FCS_SSH_EXT.1.4 Test 1 and verified that diffie-hellman-group14-sha256 kex algorithms was advertised, as claimed in the ST.
Test Result	Pass

Test Number	2

Test Objective	The evaluator verifies that the TOE rejects an SSH session establishment attempt when using the TSF-unsupported key exchange algorithm of diffie-hellman-group1-sha1.
Test Steps Performed	The evaluator configured an SSH peer to allow only the non-claimed diffie-hellman-group1-sha1 key exchange method. The evaluator verified that the connection failed.
Test Result	Pass

2.15.7 FCS_SSH_EXT.1.7

TSS

The evaluator will check the description of the implementation of SSH in the TSS to ensure the KDFs supported are specified. The evaluator will check the TSS to ensure that the KDFs specified are identical to those listed for this component.

[ST] Section 6.2.10 contains the selections for this SFR – "RFC 4253 (Section 7.2)". [ST] Section 7.2.3 specifies the KDF supported. These sections are consistent with each other.

Guidance Documentation

None.

Tests

None.

2.15.8 FCS_SSH_EXT.1.8

TSS

The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values that these lower limits are identified. In cases where hardware limitation will prevent reaching data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains:

An argument describing this hardware-based limitation and

Identification of the hardware components that form the basis of such argument.

For example, if specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.

[ST] Section 7.2.3 states, "Before either of these thresholds are reached, the TOE performs a rekey (approximately 525 mB transmitted, or 525 mB received).

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that if the connection rekey or termination limits are configurable, it contains instructions to the administrator on how to configure the relevant connection rekey or termination limits for the TOE.

[AGD] Section "SSH Rekey" describes the TSF rekey behavior for SSH.

Tests

The test harness needs to be configured so that its connection rekey or termination limits are greater than the limits supported by the TOE -- it is expected that the test harness should not be initiating the connection rekey or termination.

Test 1: Establish an SSH connection. Wait until the identified connection rekey limit is met. Observed that a connection rekey or termination is initiated. This may require traffic to periodically be sent, or connection keep alive to be set, to ensure that the connection is not closed due to an idle timeout.

Test 2: Establish an SSH connection. Transmit data from the TOE until the identified connection rekey or termination limit is met. Observe that a connection rekey or termination is initiated.

Test 3: Establish an SSH connection. Send data to the TOE until the identified connection rekey limit or termination is met. Observe that a connection rekey or termination is initiated.

Test Number	1
Test Objective	Verify that the TOE issues a session key rekey signal at the following threshold: • no more than one hour of connection time
Test Steps Performed	The evaluator configured the TOE with a 3600s rekey threshold and a 1GB data rekey threshold. The evaluator used a modified SSH client to connect to the TOE and started a timer awaiting a rekey while capturing packets. The evaluator observed that after 3 minutes had passed, the SSH client indicated a rekey had occurred.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE issues a session key rekey signal at the following threshold: • no more than one gigabyte of transmitted data
Test Steps Performed	The evaluator used a proprietary tool to execute a command repeatedly in an SSH session to receive data from the TOE. The evaluator selected a command that would return a large amount of data. The evaluator observed that when the TOE had sent 10 MB in the established session that a rekey occurred.
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE issues a session key rekey signal at the following threshold: • No more than one gigabyte of SSH data transmitted to the TOE
Test Steps Performed	The evaluator used a proprietary tool to execute a command repeatedly in an SSH session to receive data from the TOE. The evaluator selected a command that would return a large amount of data. The evaluator observed that when the TOE had sent 1 GB in the established session that a rekey occurred.
Test Result	Pass

2.16 FCS_SSHS_EXT.1 SSH Server (Selection-Based)

TSS

No activities.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

[AGD] Section “Enabling administrative access via SSH” describes the process in which to enable SSH while in the CC mode of operation.

Tests

The evaluator shall repeat Test 1 and Test 2 from FCS_SSH_EXT.1.4 for each of the authentication mechanisms supported by the TOE.
Next the evaluator shall configure the remote peer to only allow an authentication mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.

Test Number	1
Test Objective	Using the packet capture from FCS_SSH_EXT.1.4, Test 1, the evaluator verifies that the TOE proposed only rsa-sha2-256, as selected in [ST].
Test Steps Performed	The evaluator initiated a SSH connection and observed in the packet capture that the only server host key algorithm proposed by the TOE is rsa-sha2-256.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE successfully establishes an SSH session using the rsa-sha2-256 host key algorithm.
Test Steps Performed	The evaluator observed that the TOE successfully establishes an SSH session using the rsa-sha2-256 host key algorithm.
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE rejects a SSH connection attempt using the rsa-sha2-512 host key algorithm.
Test Steps Performed	The evaluator observed that the TOE established a SSH session successfully with rsa-sha2-512 host key algorithm.
Test Result	Pass

2.17 FCS_TLSC_EXT.1 Extended: TLS Client Protocol without Mutual Authentication (Selection-Based)

2.17.1 FCS_TLSC_EXT.1.1

TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.

[ST] Section 7.2.5 specifies the supported ciphersuites. [ST] Section 6.2.12 contain the selections for this SFR. These sections are consistent with each other.

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

[AGD] Section “The CC Mode of Operation” contains instructions to enable the CC mode of operation which ensures [ST] selections are enforced for the TLS client.

Tests

Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation.

Test 2: The evaluator shall establish the connection with a server presenting a certificate that contains the serverAuth (OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage extension and verify that the connection successfully negotiated. The evaluator shall then verify that when the same server presents an otherwise valid server certificate that contains the extendedKeyUsage extension without serverAuth the client rejects the connection. Ideally, the two certificates should be identical except for the OID values.

Test 3: [Conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server’s Certificate handshake message.

Test 4: The evaluator shall perform the following 'negative tests':

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the TOE TLS client denies the connection.

Modify the server’s selected ciphersuite in the Server Hello handshake message to be a ciphersuite (compatible with the server-selected version of TLS) not presented in the Client Hello handshake message. The evaluator shall verify that the TOE TLS client rejects the connection after receiving the Server Hello.

The evaluator shall attempt to establish a TLS connection using each valid TLS/SSL version (i.e. TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The evaluator shall verify that the version(s) specified in FCS_TLSC_EXT.1.1 are successfully established and all other versions are rejected by the TOE TLS client. If a supported_versions extension is not sent by the TOE in the ClientHello, then the evaluator must ensure the test server responds with a ServerHello that is valid for the TLS version being negotiated. If the TOE includes the Supported Versions extension in its ClientHello, the evaluator shall also ensure the version(s) specified in cPP_ND_v3.0e-SD, 06-Dec-2023 130 the extension match the version(s) in FCS_TLSC_EXT.1.1. NOTE: For TLS 1.3 aware test servers, it is appropriate for the test server to issue a TLS Alert. The TOE client must not attempt to continue the connection.

Test 5: The evaluator shall perform the following modifications to the traffic (i.e. Man-in-the-middle modifications that result in invalid signatures and MACs):

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. If using DHE or ECDH ciphersuites, modify the signature block in the Server's Key Exchange handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature. This test does not apply to ciphersuites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

[Conditional]: Perform this test only if support of TLS 1.3 is claimed. Modify the signature block in the Server's Certificate Verify handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature.

Test 6: The evaluator shall perform the following 'scrambled message tests':

Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows. (Note: This modification must be performed prior to the contents of the Finished message being encrypted.)

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake is not finished successfully and no application data flows. (Note: TLS 1.3 provides for a dummy ChangeCipherSpec message to aid in middlebox compatibility if such an option is enabled in the specific implementation [see Section D.4 in RFC 8446]. If TLS 1.3 middlebox compatibility mode is enabled a ChangeCipherSpec message may appear in packet traces, but it does not influence the protocol. To be clear: for TLS 1.3, this test does not need to be performed.)

[Conditional]: Perform this test only if support of TLS 1.3 is claimed. Send a plaintext EncryptedExtensions message from the server and verify that the handshake is not finished successfully and no application data flows. (Note: Under TLS 1.3, the EncryptedExtensions message is the first message to be encrypted with the handshake traffic secret.)

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

Test Number	1
Test Objective	Verify that the ciphersuite(s) claimed in the [ST] can be negotiated.
Test Steps Performed	The evaluator configured a test TLS server to restrict its supported ciphersuites to one claimed ciphersuite and verified that the TOE established a connection. The evaluator repeated this for each ciphersuite specified in the requirement.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE will reject a server certificate that does not have the Server Authentication extendedKeyUsage extension.
Test Steps Performed	The evaluator confirmed that a TLS server presenting a certificate with the Server Authentication purpose in the extendedKeyUsage field resulted in a successful connection. The evaluator then configured the TLS server with a certificate without the Server Authentication purpose in the extendedKeyUsage field, attempted a connection and verified that the TOE did not accept the connection.
Test Result	Pass

Test Number	3
Test Objective	Verify the TOE will not accept a server certificate that is not matching the server-selected ciphersuite.
Test Steps Performed	The evaluator configured the TLS server to present a server certificate that did not match the server selected ciphersuite and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection.
Test Result	Pass

Test Number	4
Test Objective	• Verify the TOE rejects TLS_NULL_WITH_NULL_NULL ciphersuite. • Verify the connection is not established when the server provides a ciphersuite not present in the Client Hello. • Verify the TOE rejects all TLS version connection attempts other than TLSv1.2 and TLSv1.3.
Test Steps Performed	The evaluator configured the TLS server to present the TLS_NULL_WITH_NULL_NULL cipher suite in the Server Hello message and attempted a connection from the TOE. The evaluator confirmed that the TOE did not accept the connection. The evaluator configured a remote TLS server to present a Server Hello message that contained a cipher suite that did not match any presented in the Client Hello handshake. The evaluator confirmed that the TOE did not accept the connection.
Test Result	Pass

Test Number	5
Test Objective	• [TOE claims TLS 1.2 and ECDHE ciphersuites] Verify the TOE rejects the connection with a corrupted signature block in the Server Key Exchange message. • [TOE claims TLS 1.3] Verify the TOE rejects the connection with a corrupted signature block in the Certificate Verify message.
Test Steps Performed	The evaluator modified the TLS server to send a Server Key Exchange message with a modified signature block. The evaluator confirmed that the TOE did not accept the connection. The evaluator configured a TLS server to modify a byte within the TLS 1.3 Certificate Verify signature. The evaluator confirmed that the TOE did not accept the connection.
Test Result	Pass

Test Number	6
Test Objective	• Verify the TOE rejects a TLS handshake with a corrupted Server Finished message (run for both TLSv1.2 and TLSv1.3), • [Condition is satisfied, TOE claims TLS 1.2] Verify the TOE rejects a TLS handshake with a garbled message before Server Finished message, • [Condition is satisfied, TOE claims TLS 1.3] Verify the TOE rejects a TLS handshake with plaintext EncryptedExtensions message. • [Condition is satisfied, TOE claims TLS 1.2] Verify the TOE rejects a TLS handshake with a modified nonce in the server key exchange message.

Test Steps Performed	The evaluator modified a byte in the Server Finished record before encrypting and sending it to the client and observed the TOE terminating the connection after receiving the Server Finished message. The evaluator configured a TLS server to send a garbled application data message instead of a Finished record after the ChangeCipherSpec message and confirmed the TOE rejected the connection. The evaluator configured a TLS server to modify the nonce sent in the Server Hello handshake message and confirmed the TOE terminated the connection after receiving the Server Key Exchange message.
Test Result	Pass

2.17.2 FCS_TLSC_EXT.1.2

TSS

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application- configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS shall describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

[ST] Section 7.2.5 describes the client's method of establishing all reference identifiers, with the following details:

- SAN:
  - IPv4 and IPv6 addresses
  - DNS names
  - Wildcard supported
- CN:
  - DNS names
  - Wildcard supported

Guidance Documentation

The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

[AGD] Section titled "Reference Identifiers" provides guidance on reference identifiers. The guidance is consistent with the description of reference identifiers in [ST] Section 7.2.5.

Tests

Note that the following tests are marked conditional and are applicable under the following conditions:

For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

Note that for some tests additional conditions apply.

IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.

IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI- ID):

[conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo..example.com) and verify that the connection fails.*

*[conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)*

Test 6: Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

[conditional]: If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=.168.0.1 when connecting to 192.168.0.1, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6).*

Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modifiy each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at- serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of

the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.

The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

Test Number	1
Test Objective	Condition is satisfied by the TOE claims] For the following reference identifier types, verify that the TOE rejects the TLS handshake when a server certificate is presented that contains a CN that does not match the reference identifier and does not contain the SAN extension: • CN-ID (FQDN in the CN)
Test Steps Performed	The evaluator configured the TLS server to present a certificate with a CN that does match the reference identifier and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. For this test, the evaluator generated a certificate with an invalid FQDN.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE rejects a TLS handshake that presents a server certificate with a good CN but a bad SAN, repeated using the following reference identifier types supported by the TOE: • IPv4 in the SAN • IPv6 in the SAN • FQDN in the SAN
Test Steps Performed	The evaluator configured the TLS server to present a certificate with a CN that does match the reference identifier and a SAN extension with a value that does not match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE did not accept the connection. The evaluator tested a bad IPv4, IPv6 and FQDN in the SAN extension.
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE accepts a TLS connection when sending a server certificate with a good CN, repeated for the following reference identifier types supported by the TOE: • FQDN in the CN
Test Steps Performed	The evaluator configured the TLS server to present a certificate with a valid CN and no SAN extension. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN in the CN field.
Test Result	Pass

Test Number	4
Test Objective	Verify that the TOE accepts a TLS connection when sending a server certificate with a bad CN but a good SAN, repeated for the following reference identifier types supported by the TOE: • IPv4 in the SAN • IPv6 in the SAN • FQDN in the SAN
Test Steps Performed	The evaluator configured the TLS server to present a certificate with a CN that does not match the reference identifier and a SAN extension with a value that does match the reference identifier. The evaluator attempted a connection from the TOE and verified that the TOE accepted the connection. The evaluator tested FQDN in the SAN extension.
Test Result	Pass

Test Number	5
Test Objective	Verify that the TOE rejects a TLS handshake when the server cert contains the following: • a wildcard not in the left-most label (CN-ID: FQDN in CN) • a wildcard not in the left-most label (DNS-ID: FQDN in SAN) Verify that the TOE performs the following with the following configurations: • TOE establishes a TLS session when the presented server cert contains a wildcard label in the leftmost DNS label (*.example.com) of the CN with no SAN present. • TOE establishes a TLS session when the presented server cert contains a wildcard label in the left-most DNS label (*.example.com) of the SAN. Verify that the TOE rejects connections to the server with certificate with wildcard in left-most label with the following configurations: • TOE is configured with domain name that is missing the leftmost label and presented server cert has left-most wildcard in the in the CN and no SAN. • TOE is configured with domain name that is missing the leftmost label and presented server cert has left-most wildcard in the in the SAN. • TOE is configured with extra left-most label and presented server cert has left-most wildcard in the in the CN and no SAN. • TOE is configured with extra left-most label and presented server cert has left-most wildcard in the SAN.
Test Steps Performed	• The evaluator configured the www.*.local in the CN and SAN field and observed that the connection got rejected. • The evaluator configured *.server.local in the CN and SAN field and observed that the connection is established. • The evaluator configured *.www.server.local in the CN and SAN of the leaf certificate is and observed that connection is not established. • The evaluator configured bar.www.server.local in the CN and SAN of the leaf certificate and observed that connection is not established.
Test Result	Pass

Test Number	6
Test Objective	Verify the TOE supports a certificate with a wildcard in the left-most label in the CN if claimed in the [ST], otherwise confirm that TOE does not support wildcards. [ST] states TSF [does or does not support wildcards].
Test Steps Performed	This test is not applicable as the TOE doesn't support wildcards in IP Addresses.
Test Result	Pass

Test Number	7
Test Objective	This test is not applicable because the TOE is not distributed and does not claim FPT_ITT.1.
Test Steps Performed	N/A
Test Result	Pass

2.17.3 FCS_TLSC_EXT.1.3

TSS

None.

Guidance Documentation

None.

Tests

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Test 2: [Conditional]: If "except with the following administrator override" is selected, the evaluator shall change the presented certificate(s) or modify the operational environment, so that certificate validation fails due to the TSF's inability to determine revocation status. The evaluator shall verify that the certificate is not accepted by the TSF until the Security Administrator authorizes the TSF to establish the connection and this action results in the Trusted Channel being successfully established.

Test 3 [conditional]: While performing testing of invalid TLS Client Reference Identifiers, expired X.509 certificates, and invalid X.509 trust chains; the evaluator shall ensure the TSF does not present an administrator override option, with the expcetion of failure to determine revocation status (if selected). Note: This should be a review of behiavor observed while performing other tests.

Test Number	1
Test Objective	Verify that the TOE creates a trust chain of certificates and can authenticate and establish a trusted channel.
Test Steps Performed	Initially, the evaluator configured a TLS server to present a certificate that did not chain back to a certificate in the TOE's trust store. The evaluator observed that the connection failed. The evaluator then loaded into the TOE's trust store the root CA certificate and intermediate CA certificates needed to validate the TLS server's certificate and attempted a connection

	from the TOE to the TLS server. The evaluator confirmed that the connection succeeded.
Test Result	Pass

Test Number	2
Test Objective	This test is not applicable as the TOE doesn't support administrator override mechanism.
Test Steps Performed	N/A
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE doesn't provide an administrator override mechanism while performing testing for invalid TLS client reference identifiers, expired X509 certificates and invalid X.509 trust chains.
Test Steps Performed	The evaluator observed while configuring the TLS Server on the TOE that there was no administrator override mechanism present.
Test Result	Pass

2.17.4 FCS_TLSC_EXT.1.4

TSS

If "present the Supported Groups Extension" is selected, the evaluator shall verify that TSS describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured. If TLS 1.2 is claimed and DHE ciphers are claimed, then the TSS must also specify whether the TOE is capable of negotiating DHE ciphers and whether the TOE client will terminate if an unsupported DHE parameter set is returned in the Server Key Exchange or whether all valid server-generated DHE parameters are accepted.

[ST] Section 7.2.5 describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured.
TLS 1.2 is claimed and DHE ciphers are claimed: [ST] Section 7.2.5 specifies that the TOE is capable of negotiating DHE ciphers. [ST] Section 7.2.5 states "When the TOE is a TLS server, it also permits MODP sizes of 2048-bits when an DHE cipher suite is negotiated."

Guidance Documentation

If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

[ST] TSS does not state that the Support Groups must be configured to fit the requirement, other than being in the CC mode of operation.

Tests

Test 1 [conditional]: If "not present the Supported Groups Extension" is selected, the evaluator shall examine the Client Hello message and verify it does not contain the Supported Groups extension.

Test 2 [conditional]: If "present the Supported Groups Extension" is selected, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE's supported groups. The evaluator shall verify that the connection succeeds. This test shall be repeated for each type

of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

Test 3 [conditional]: If secp curves are selected, the evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve and shall verify that the connection fails and no application data flows. The non-supported curve shall be as similar to the selected curve(s) as possible (i.e. a non-selected curve when not all curves are selected or P-224). This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).

Test 4a [conditional, for TLS 1.3 only]: If ffdhe curves are selected, the evaluator shall configure the server to perform a DHE key exchange in the TLS connection using a non-supported group and shall verify that the connection fails and no application data flows. The non-supported group shall be as similar to the selected group(s) as possible (i.e. a non-selected group when not all groups are selected or undefined Codepoint 0x0105 (ffdhe8192 + 1)).

Test 4b [conditional, for TLS 1.2 only]: If ffdhe curves are selected, the evaluator shall configure the server to return DHE parameters in the Server Key Exchange in the TLS connection that do not meet the construction for any claimed ffdhe group. The evaluator shall verify that the connection fails and no application data flows. If the TOE client supports any server-returned DHE parameter set, then this test is not applicable.

Test Number	1
Test Objective	This test is not applicable as the TOE supports supported Groups Extension.
Test Steps Performed	N/A
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE is able to establish sessions with the groups claimed in the ST.
Test Steps Performed	The evaluator configured a test TLS server to allow only one supported group and verified that the TOE could establish a connection successfully. The evaluator repeated this test for each claimed key exchange group (secp256r1, secp384r1, secp521r1) and supported version of TLS (TLS v1.2, TLS v1.3).
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE does not support unsupported curve for TLS1.2. Verify that the TOE does not support unsupported curve for TLS1.3.
Test Steps Performed	The evaluator configured a TLS server to negotiate only the unsupported secp224r1 and x25519 curve. The evaluator had the TOE attempt to connect to the server and verified that this connection failed. The evaluator performed this test for both TLS v1.2 and TLS v1.3.
Test Result	Pass

Test Number	4a
Test Objective	This test is not applicable as the TOE doesn't support ffdhe curves.
Test Steps Performed	N/A

Test Result	Pass

Test Number	4b
Test Objective	This test is not applicable as the TOE doesn't support ffdhe curves.
Test Steps Performed	N/A
Test Result	Pass

2.17.5 FCS_TLSC_EXT.1.5

TSS

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms extension and whether the required behavior is performed by default or may be configured.

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms_cert extension and whether the required behavior is performed by default or may be configured.

[ST] Section 7.2.5 states that the signature_algorithms extension is supported.

Guidance Documentation

If the TSS indicates that the signature_algorithms extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithms extension.

[ST] TSS does not state that the Support Groups must be configured to fit the requirement, other than being in the CC mode of operation.

Tests

Test 1 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms extension" is selected:

The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.

The evaluator shall establish a TLS connection using each of the SignatureSchemes specified by the requirement and observes the session is successfully completed. The evaluator shall ensure the test server sends a leaf Certificate that has a public key algorithm that is consistent with the SignatureScheme being tested. For TLS 1.2 and if the ciphersuite is DHE or ECDHE, the evaluator shall ensure that the server sends Server Key Exchange messages consistent with the SignatureScheme being tested. For TLS 1.3, the evaluator shall ensure that the server sends Certificate Verify messages consistent with the SignatureScheme being tested.

Test 2 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms_cert extension" is selected:

The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms_cert extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.

The evaluator shall establish a TLS connection using a certificate chain using each of the SignatureSchemes specified by the requirement. The evaluator shall ensure the signatures used in the certificate chain are consistent with the SignatureScheme being tested.

Test Number	1
Test Objective	Verify that the TOE doesn't support any unsupported signature schemes. Verify that the TOE is able to establish TLS sessions with the following signature algorithms:
Test Steps Performed	The evaluator configured a TLS server to force the use of only one signature algorithm. The evaluator verified that the TOE successfully connected using each supported signature algorithm.
Test Result	Pass

Test Number	2
Test Objective	This test is not applicable as the TOE doesn't support signature_algorithms_cert extension.
Test Steps Performed	N/A
Test Result	Pass

2.17.6 FCS_TLSC_EXT.1.6

TSS

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

[ST] Section 7.2.5 states that the list of supported ciphersuites can not be configured, with the following details:

- "The TOE operates in FIPS-CC mode of operation that restricts the cipher suites and algorithms used by HTTPS/TLS to those identified above."

Guidance Documentation

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

[AGD] Section "Web browser requirements" lists the TLS ciphersuites supported in the CC mode of operation. The TLS ciphersuites are not configurable per [ST] FCS_TLSC_EXT.1.6 and FCS_TLSS_EXT.1.5 selections.

Tests

Test 1 [conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

Test Number	1
Test Objective	This test is not applicable as the TOE doesn't provide the ability to configure ciphersuites.
Test Steps Performed	N/A
Test Result	Pass

2.17.7 FCS_TLSC_EXT.1.7

TSS

None.

Guidance Documentation

None.

Tests

Test 1: The evaluator shall establish a TLS connection with a server and observe that the early data extension and the post-handshake client authentication extension according to RFC 8446 Section 4.2 are not advertised in the Client Hello Message. This test shall be executed for all TLS versions supported by the TOE.

Test Number	1
Test Objective	Verify that the TLS1.3 session established by the TOE doesn't advertise the early data extension and post handshake client authentication.
Test Steps Performed	The evaluator had the TOE connect to a compliant TLS server while capturing packets. The evaluator observed the packet trace and verified that the TOE did not send the early_data or post_handshake_auth extensions in the Client Hello. The evaluator performed this test for TLS v1.2 and TLS v1.3.
Test Result	Pass

2.17.8 FCS_TLSC_EXT.1.8

TSS

The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

For TLS 1.3: [ST] Section 7.2.5 describes the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

Guidance Documentation

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

[AGD] Section "The CC Mode of Operation" describes enabling the CC mode of operation as the configuration necessary to limit ciphersuites to those selected in the [ST]. Per [ST], the TSF does not support PSKs outside of TLSv1.3 session resumption.

Tests

None.

2.17.9 FCS_TLSC_EXT.1.9

TSS

None.

Guidance Documentation

None.

Tests

The evaluator shall perform the following tests: *Test 1 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that either the "renegotiation_info" field or the SCSV ciphersuite is included in the ClientHello message during the initial handshake.* *Test 2 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify the TOE TLS Client's handling of ServerHello messages received during the initial handshake that include the "renegotiation_info" extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the TOE TLS client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.* *Test 3 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify that ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. The evaluator shall modify either the "client_verify_data" or "server_verify_data" value and verify that the TOE TLS client terminates the connection.* *Test 4a [conditional, if the TOE supports TLS 1.3]: The evaluator shall initiate a TLS session between the TSF and a test TLS 1.3 server that completes a compliant TLS 1.3 handshake, followed by a hello request message. The evaluator shall observe that the TSF completes the initial TLS 1.3 handshake successfully, but terminates the session on receiving the hello request message.* *It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).* *Test 4b [conditional, if the TOE supports TLS 1.2 and rejects TLS 1.2 renegotiation attempts]: The evaluator shall initiate a TLS session between the so-configured TSF and a test TLS 1.2 server that is configured to perform a compliant handshake, followed by a hello request. The evaluator shall confirm that the TSF completes the initial handshake successfully but does not initiate renegotiation after receiving the hello request.*

Test Number	1
Test Objective	This test is not applicable as the TOE does not support secure renegotiation.
Test Steps Performed	N/A
Test Result	Pass

Test Number	2
Test Objective	This test is not applicable as the TOE doesn't support secure renegotiation for TLS1.2.
Test Steps Performed	N/A
Test Result	Pass

Test Number	3

Test Objective	This test is not applicable as the TOE doesn't support secure renegotiation for TLS1.2.
Test Steps Performed	N/A
Test Result	Pass

Test Number	4a
Test Objective	Verify that the TOE rejects renegotiation attempts made by the server for TLS1.3.
Test Steps Performed	The evaluator configured a TLS server to establish a handshake and attempt to renegotiate. The evaluator observed that the TOE refused to renegotiate the connection and remained on the existing negotiated parameters and session. The evaluator performed this test for TLS 1.3 connections.
Test Result	Pass

Test Number	4b
Test Objective	Verify that the TOE rejects renegotiation attempts made by the server for TLS1.2.
Test Steps Performed	The evaluator configured a TLS server to establish a handshake and attempt to renegotiate. The evaluator observed that the TOE refused to renegotiate the connection and remained on the existing negotiated parameters and session. The evaluator performed this test for both TLS 1.2 connections.
Test Result	Pass

2.18 FCS_TLSS_EXT.1 Extended: TLS Server Protocol without Mutual Authentication (Selection-Based)

2.18.1 FCS_TLSS_EXT.1.1

TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of unsupported and undefined SSL and TLS versions.

[ST] Section 7.2.5 specifies the supported ciphersuites. [ST] Section 6.2.12 contains the selections for this SFR. These sections are consistent with each other.

[ST] Section 7.2.5 describes how the TOE technically prevents the use of unsupported and undefined SSL and TLS versions, with the following details:

- "The TOE implements TLS 1.2 and TLS 1.3 as both a client and a server, rejecting all other TLS/SSL versions. Specifically, enabling FIPS-CC mode (required in evaluated configuration) forces the TOE to use only TLS versions 1.2 or 1.3. All other versions are automatically disabled by default."

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE or TLS version supported by the TOE may have to be restricted to meet the requirements).

[AGD] Section “The CC Mode of Operation” describes enabling the CC mode of operation as the only configuration needed to restrict ciphersuites with [ST] selected ciphersuites.

Tests

Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

Test 2: The evaluator shall perform the following tests:

The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server’s ST and verify that the server denies the connection.

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

Test 3: The evaluator shall perform the following modifications to the traffic:

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

(The intent of this test is to ensure that the server’s TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt TLS Finished message and b) Encrypt every TLS message after session keys are negotiated.)

[Conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent. The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server’s ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55…) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c…), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

[Conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing a single curve in the Supported Groups extension. The curve that is selected to be presented in this extension should not be supported by the TOE. The evaluator shall verify that the TOE disconnects after receiving the Client Hello message.

[Conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing multiple curves in the Supported Groups extension. These curves should

be chosen such that only one of these curves is supported by the TOE. The evaluator shall verify that the TOE responds with a Hello Retry Request message selecting the supported curve. This shall be reflected in the Key Share extension of the Hello Retry Request message.

Test 4: The evaluator shall attempt to establish a TLS/SSL connection using each of the supported TLS/SSL versions (i.e., TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The client shall be configured so it only supports the version being tested. The evaluator shall verify that the versions specified in FCS_TLSS_EXT.1.1 are successfully established and all other versions not successfully established. If the TOE attempts to downgrade the version, it is acceptable for the test client to terminate the connection; however, the version selected by the TOE shall always be a version specified in FCS_TLSS_EXT.1.1.

Test Number	1
Test Objective	Verify the ciphersuite(s) claimed in [ST] can be negotiated.
Test Steps Performed	The evaluator configured a TLS client to request each of the cipher suites claimed in [ST] and attempted a connection to the TOE. The evaluator confirmed that negotiation of each cipher suite was successful.
Test Result	Pass

Test Number	2
Test Objective	• Verify the TOE denies a connection when using an unsupported cipher suite. • Verify that the TOE denies a connection with TLS_NULL_WITH_NULL_NULL cipher suite.
Test Steps Performed	The evaluator configured a TLS client to request a list of cipher suites not supported by the TOE (based on the list of claimed cipher suites in the ST) and attempted a connection to the TOE. The evaluator confirmed that the TOE did not accept the connection. Next, the evaluator configured a TLS client to request the TLS_NULL_WITH_NULL_NULL cipher suite and attempted a connection to the TOE. The evaluator confirmed that the TOE did not accept the connection
Test Result	Pass

Test Number	3a
Test Objective	Verify that the TOE terminates the connection with a modified client finished handshake message and doesn't send any encrypted data.
Test Steps Performed	The evaluator configured a TLS client to modify the last byte in the Client Finished handshake message and sent the modified Client Finished message to the TOE. The evaluator observed that the TOE terminated the connection and did not send any application data.
Test Result	Pass

Test Number	3b
Test Objective	Verify that the TOE performs successful TLS1.2 handshake with transmission of encrypted application data.
Test Steps Performed	The evaluator used a TLS client to carry out a compliant handshake with the TOE and to send application data while capturing packets. It was verified that the Server's Encrypted Handshake Message was truly encrypted, and no Alert messages were sent. The evaluator examined the frame number 6 and found that the data bytes for Encrypted Handshake Message did not contain "16 03 03 00 40 14 00 00 0c" but were truly encrypted.
Test Result	Pass

Test Number	3c
Test Objective	Verify that the TOE disconnects a TLS1.3 connection after Client Hello after encountering an unsupported curve.
Test Steps Performed	The evaluator configured a TLS client to send a Client Hello containing only an unsupported curve. The evaluator verified that the TOE rejected the handshake.
Test Result	Pass

Test Number	3d
Test Objective	Verify that the TOE selects the supported curve (secp384r1) from the Client Hello to establish a successful TLS1.3 connection.
Test Steps Performed	The evaluator configured a TLS client to send a list of multiple curves to the TOE, only one of which was supported by the TOE. The evaluator verified that the TOE responded with a Hello Retry Request selecting the supported curve in the key_share extension.
Test Result	Pass

Test Number	4
Test Objective	• Verify the TOE rejects all connection attempts to all TLS/SSL versions (i.e.TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). • Verify that the TOE accepts connections from the supported version, [TLS1.2 and TLS1.3].
Test Steps Performed	The evaluator configured a TLS client to attempt a connection with versions of TLS not supported by the TOE (i.e., SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1). The evaluator observed that the TOE rejected all connection attempts. The ability of the TOE to accept and successfully complete connection attempts using TLS v1.2 and TLS v1.3 was demonstrated in FCS_TLSS_EXT.1.1 Test 1. In that test, the evaluator iterated through all TOE-supported cipher suites across TLS 1.2 and TLS 1.3 and verified that the TOE accepted connections with these versions.
Test Result	Pass

2.18.2 FCS_TLSS_EXT.1.2

TSS

The evaluator shall verify that the TSS describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

[ST] Section 7.2.5 describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients, with the following details:

- "The TOE supports server authentication via x.509v3 certificates using RSA keys of 2048 bits, or ECDSA key sizes of 256 bits, 384 bits, or 512 bits."

[ST] Section 6.2.13 contains the selections for this SFR. These sections are consistent with each other.

Guidance Documentation

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

[AGD] Section “The CC Mode of Operation” describes enabling the CC mode of operation as the only configuration needed.

Tests

Test 1: [conditional]: If ECDHE ciphersuites/group are supported:

The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite (TLS 1.2) or group (TLS 1.3) and a single supported elliptic curve specified in the supported groups extension. The Evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange (TLS 1.2) or Server Hello (key_share, for TLS 1.3) message and successfully establishes the connection.

For TLS 1.2, the evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g., secp192r1 (0x13)) specified in RFC 4492, Section 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

For TLS 1.3, the evaluator shall attempt a connection using a supported ciphersuite and a single unsupported group. Both the key_share and supported_groups extensions must be set to the same unsupported group. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

Test 2 [conditional]: If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite.

For TLS 1.2, the evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

For TLS 1.3, the evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Share Extension Message where the KeyShareServerHello structure contains a KeyShareEntry structure with an opaque key_exchange value whose Length is consistent with the configured Diffie-Hellman parameter size(s).

Test 3: [Conditional]: If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

Test Number	1
Test Objective	a) Verify the TOE establishes the connection using the curves claimed in the ST for supported TLS versions. b) Verify that the TOE rejects a TLS1.2 client connection when established using the unsupported (x25519) curve. c) Verify that the TOE rejects a TLS1.3 client connection when established using the unsupported (x25519) curve.
Test Steps Performed	The evaluator configured a TLS client to support the given version of TLS with a supported ECDHE ciphersuite/group and verified that each supported curve could be used to establish a connection.

	The evaluator configured the TLS client to send a supported TLS 1.2 ECDHE cipher suite with an unsupported elliptic curve and verified that the TOE rejected the connection. The evaluator configured a TLS client to send only a group that was not supported by the TOE and verified that the TOE rejected the connection.
Test Result	Pass

Test Number	2
Test Objective	[ST] claims DH parameter support for only MODP 2048. Verify that the TOE sends a Server Key Exchange message with the parameter length specified in the ST while establishing TLSv1.2 connections.
Test Steps Performed	The evaluator initiated a TLS client connection to the TOE and observed in the packet capture that the DH parameter (p-value) value is 256 bytes, indicating that MODP 2048 was used. [ST] does not claim support for ffdhe groups in TLS1.3.
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE is able to establish TLS1.2 connections with every RSA ciphersuite for key establishment as mentioned in the ST.
Test Steps Performed	This test is not applicable as the TOE doesn't support any specific RSA key establishment ciphersuites.
Test Result	Pass

2.18.3 FCS_TLSS_EXT.1.3

TSS

The evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

[ST] Section 7.2.5 lists all EC Diffie-Hellman curves and Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server, with the following details:

- "The TLS client and server uses secp256r1, secp384r1, or secp521r1 when a TLS_ECDHE cipher suite is negotiated.

[ST] Section 6.2.13 contains the selections for this SFR. These sections are consistent with each other.

Guidance Documentation

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

[AGD] Section "The CC Mode of Operation" describes enabling the CC mode of operation as the only configuration needed.

Tests

The requirement for this SFR is completed in the 'Tests' portion for FCS_TLSS_EXT.1.2.

2.18.4 FCS_TLSS_EXT.1.4

TSS

The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246), if session resumption based on session tickets is supported (RFC 5077) and/or if session resumption according to RFC 8446 is supported.

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

If the TOE claims a TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator shall verify that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used, the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

[ST] Section 7.2.5 states that session resumption based on session tickets is supported for TLSv1.2 and TLSv1.3 with the following details:

- "The TLS server supports session resumption based on session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077 for TLSv1.2 and RFC 8446 for TLSv1.3."

[ST] Section 7.2.5 states that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption, with the following details:

- "Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm consistent with FCS_COP.1/DataEncryption. Depending on the negotiated ciphersuite, AES in CBC or GCM mode (128, 256 bits) are used to protect session tickets."

[ST] Section 7.2.5 states that the session tickets "adhere to the structural format provided in section 4 of RFC 5077 for TLSv1.2 and RFC 8446 for TLSv1.3.

Guidance Documentation

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

[AGD] Section "The CC Mode of Operation" describes enabling the CC mode of operation as the only configuration needed.

Tests

Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).

Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC 5246 (TLS 1.2) or session tickets according to RFC 5077 (TLS 1.2) or session resumption according to RFC 8446 (TLS 1.3), the evaluator shall perform the following test:

For all supported TLS versions the client shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. A non-zero length session identifier for TLS 1.3 would result in testing compatibility mode which is not the objective of this test. For TLS 1.3, the evaluator shall ensure that a 'psk_key_exchange_modes' extension is included in the Client Hello.

The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

The client verifies the Server Hello message contains a zero-length session identifier. For TLS 1.2 the client could alternatively pass the following steps (not applicable for TLS 1.3):

Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

The client completes the TLS handshake and captures the SessionID from the ServerHello.

The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1. [Conditional]: If the TOE supports session resumption using session IDs according to RFC 5246 (TLS 1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Test 3 [Conditional]: If the TOE supports session tickets according to RFC5077 (supported only by TLS 1.2), the evaluator shall carry out the following steps:

The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in Section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in Section 3.3 of RFC 5077. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.
Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Test 4 [Conditional]: If the TOE supports session resumption according to RFC8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps:

The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the pre-shared key in the ClientHello. The evaluator shall confirm that the TOE responds similarly to figure 3 of RFC 8446 after successfully reusing the pre-shared-key to resume the session. Specifically, the server must not send back a Certificate message if the session is correctly resumed. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.

The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the pre-shared key and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.

The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then force the non-TOE client to attempt to establish a new connection using the previous session ticket material as a pre-shared key, but set psk_key_exchange_modes with a value of psk_ke in the Client Hello message and omit the psk_ke_dhe. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.

Test Number	1
Test Objective	• For TLS1.2, verify that the TOE doesn't support session IDs or session tickets by sending a zero length session ID or session ticket and observe that the TOE implicitly rejects the previous session ID or session ticket by performing a full handshake or terminates the connection. • For TLS1.3, verify that the TOE doesn't support session resumption by using the previous session's PSK and observe the

	TOE terminating the connection or implicitly rejecting the PSK and performing a full handshake.
Test Steps Performed	The TOE supports session resumption using RFC5077 for TLS1.2 and RFC8446 for TLS1.3, hence this test is not applicable.
Test Result	Pass

Test Number	2
Test Objective	• Verify that the TOE establishes an abbreviated TLS handshake when a session is resumed using a previous successful TLS session's session ID. • Verify that the TOE terminates a TLS connection or implicitly performs a complete handshake when a session is resumed using a session ID of a previous TLS handshake that was terminated by generating an alert.
Test Steps Performed	The TOE supports session resumption according to RFC 5077 for TLS1.2, hence this test is not applicable.
Test Result	Pass

Test Number	3
Test Objective	• Verify that the TOE resumes a session with an abbreviated handshake when a client attempts to reuse the previous session's ticket. • Verify that the TOE drops or implicitly rejects a TLS connection and performs complete handshake established when the client tries to use a previous session ticket which has been modified.
Test Steps Performed	The evaluator configured a TLS client to attempt session resumption via a session ticket. The evaluator verified that the TOE accepted the ticket and resumed the session. The evaluator configured a TLS client to establish a connection and attempt to resume with a modified session ticket. The evaluator verified that the TOE rejected the ticket and initiated a new TLS session handshake.
Test Result	Pass

Test Number	4i
Test Objective	Verify that the TOE performs a successful TLS1.3 abbreviated handshake, when a pre-shared key of a previous successful TLS1.3 session is used.
Test Steps Performed	The evaluator configured a TLS client to complete a TLS handshake and then attempted to correctly reuse the session. The evaluator observed that the session was resumed. The evaluator determined the TOE did not send a Certificate message when correctly resuming the session. The evaluator additionally observed that the server did not advertise the early data extension.
Test Result	Pass

Test Number	4ii
Test Objective	Verify that the TOE drops/implicitly rejects the modified PSK of a previous session and performs a full TLS1.3 handshake.
Test Steps Performed	The evaluator configured a TLS client to complete a TLS handshake and attempted to resume the session but modify the pre-shared key first. The

	evaluator observed that the TOE rejected the session ticket and terminated the connection.
Test Result	Pass

Test Number	4iii
Test Objective	Verify that the TOE drops/ implicitly rejects a TLS1.3 connection where a previous successful TLS1.3 session's PSK is used where the key exchange mode is modified.
Test Steps Performed	The evaluator configured a TLS client to complete a TLS handshake and attempted to resume the session with the correct PSK but omitted the psk_ke_dhe value within psk_key_exchange_modes. The evaluator observed that the TOE terminated the session when it received the ticket.
Test Result	Pass

2.18.5 FCS_TLSS_EXT.1.5

TSS

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

[ST] Section 7.2.5 states that the list of supported ciphersuites cannot be configured, with the following details:

- "The TOE operates in FIPS-CC mode of operation that restricts the cipher suites and algorithms used by HTTPS/TLS to those identified above. The administrator does not need to take any specific actions to ensure compliance once the FIPS-CC mode of operation has been enabled."

Guidance Documentation

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

[ST] states the TSF does not provide the ability to configure the list of supported ciphersuites, as made clear in [AGD] Section "Remote access requirements".

Tests

Test 1 [conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.

Test Number	1
Test Objective	Verify that the TOE can configure the list of supported ciphersuites and establish successful TLS sessions with each of them.
Test Steps Performed	This test is not supported as the TOE doesn't provide the evaluator with the ability to configure the supported ciphersuites.
Test Result	Pass

2.18.6 FCS_TLSS_EXT.1.6

TSS

None.

Guidance Documentation

None.

Tests

Test 1: According to RFC 8446 Section 4.2.10, a PSK is required to use the early data extension. As NDcPP only allows the use of PSK in conjunction with session resumption, a NDcPP conformant TOE which acts as TLS Server cannot use the early data extension if session resumption is not supported. For TOEs that do not support session resumption, execution of test FCS_TLSS_EXT.1.4 Test 1 is regarded as sufficient that the TOE does not support the early data extension. For TOEs that support session resumption, FCS_TLSS_EXT.1.4 Test 2(i), 3(i) or 4(i) (depending on the supported TLS versions and the way session resumption is implemented) ensure that the TOE does not support the early data extension.

Test Number	1
Test Objective	Per the Test Activity, execution of FCS_TLSS_EXT.3(i) and 4(i) satisfy this requirement.
Test Steps Performed	N/A
Test Result	Pass

2.18.7 FCS_TLSS_EXT.1.7

TSS

The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

[ST] Section 6.2.13 selected “not use PSKs” for FCS_TLSS_EXT.1.7, thus, no statement needs to be made in the TSS with regards to out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

Guidance Documentation

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

[AGD] Section “The CC Mode of Operation” describes enabling the CC mode of operation as the only configuration needed.

Tests

None.

2.18.8 FCS_TLSS_EXT.1.8

TSS

None.

Guidance Documentation

None.

Tests

Test 1 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that the "renegotiation_info" extension is included in the ServerHello message. *Test 2 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero. The evaluator shall verify that the TOE TLS server sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.* *Test 3 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation. The evaluator shall verify that the TOE TLS server terminates the connection.* *Test 4a [conditional, if the TOE supports TLS 1.3]: The evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate TLS 1.3. The evaluator shall initiate a valid initial TLS 1.3 session, send a valid client hello on the non-renegotiable TLS channel, and observe that the TSF terminates the session.* *Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert).* *Test 4b [conditional, if the TOE supports TLS 1.2 and rejects TLS 1.2 renegotiation attempts]: The evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate TLS 1.2 and reject renegotiation. The evaluator shall initiate a valid initial TLS 1.2 session, send a valid ClientHello on the non-renegotiable TLS channel, and observe that the TSF does not perform renegotiation of the TLS channel.*

Test Number	1
Test Objective	Verify that renegotiation info is present in Server Hello when a session is successfully renegotiated by the TOE.
Test Steps Performed	This test is not supported as the TOE doesn't renegotiate for both TLS1.2 and TLS1.3.
Test Result	Pass

Test Number	2
Test Objective	Verify that the TOE terminates a TLS connection when the length portion of the Client Hello is modified to be non-zero.
Test Steps Performed	This test is not supported as the TOE doesn't renegotiate for both TLS1.2 and TLS1.3.
Test Result	Pass

Test Number	3
Test Objective	Verify that the TOE terminates a TLS connection when the client_verify_data or server_verify_data is modified in the Client Hello message received during renegotiation.
Test Steps Performed	This test is not supported as the TOE doesn't renegotiate for both TLS1.2 and TLS1.3.

Test Result	Pass

Test Number	4a
Test Objective	Verify that the TOE rejects the renegotiation attempt made after establishing a successful TLS1.3 handshake.
Test Steps Performed	The evaluator configured a TLS client to establish a valid TLS session and then initiated a renegotiation. After receiving the renegotiation attempt, the TOE sent an alert indicating that it does not support renegotiation and terminated the connection.
Test Result	Pass

Test Number	4b
Test Objective	Verify that the TOE rejects the renegotiation attempt made after establishing a successful TLS1.2 handshake.
Test Steps Performed	The evaluator configured a TLS client to establish a valid TLS session and then initiated a renegotiation. After receiving the renegotiation attempt, the TOE sent an alert indicating that it does not support renegotiation and terminated the connection.
Test Result	Pass

2.19 FIA_AFL.1 Authentication Failure Management

TSS

The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

[ST] Section 7.3.1 contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked, with the following details:

- “The TOE provides administrators with the capability to specify a maximum number of authentication attempts between 3 and 20 (default 4) that can be attempted before a user account is locked out from the remote GUI or SSH.”
- “The TSF increments the failure counter in non-volatile memory so that the accumulated number of failures is persisted across reboots. The failure counter is per administrator account.”

[ST] Section 7.3.1 describes the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability, with the following details:

- “The TOE provides administrators with the capability to specify a maximum number of authentication attempts between 3 and 20 (default 4) that can be attempted before a user account is locked out from the remote GUI or SSH.”
- “In the event the remote administrator is locked out, access is restored by waiting for the lockout period to expire. The lockout period is a configurable value between 1 and 60 minutes, with a default value of 3 minutes. Once the maximum number of attempts has been reached, the account will become locked and inaccessible until an administrator configured period of time has been met.”

[ST] Section 7.3.1 states that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking), with the following details:

- "The TOE supports a local interface that is not subject to the authentication failure lockout function and does not lock an administrative user out. Even if an administrator is locked out from a remote interface, they can still login locally."
- "For a virtual TOE, the administrator signs in to ESXi console to launch a local console for each VM. This prevents a situation where no administrator access is available."

Guidance Documentation

The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

[AGD] Section "Miscellaneous administration related changes" mentions the default lockout after unsuccessful login attempts and notes that these can be configured. This section provides guidance on configuring both the threshold for lockout and the time of lock out period.

Tests

The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application):

a) Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

b) Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows.

   If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator).

   If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

Test Number	1

Test Objective	Verify the TOE does not grant access after a set number of failed authentication attempts (between 3 and 20).
Test Steps Performed	The evaluator configured the TOE's number of successive unsuccessful authentication attempts to 4 and the lockout period to 5 minutes. The evaluator then attempted to authenticate the TOE with incorrect credentials 4 times, and on the 5th time attempted to use the correct credentials. The evaluator confirmed that the TOE did not allow access. This test was performed for the web GUI.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE does not allow access within a set lockout time after exceeding set authentication login failures.
Test Steps Performed	The ST includes only the time period selection in FIA_AFL.1.2. After the TOE locked the user account in Test 1 above, the evaluator started a timer. The evaluator attempted to log in by using the correct credentials twice before the lockout duration expired: the first time was 2 minutes before the lockout duration and the second time was just before the lockout expired. The evaluator verified that both attempts failed. The evaluator waited until just after the lockout expired and attempted to login using the correct credentials. The evaluator verified that this attempt was successful.
Test Result	Pass

2.20 FIA_PMG_EXT.1 Password Management

TSS

The evaluator shall check that the TSS:

lists the supported special character(s) for the composition of administrator passwords.

to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

[[ST] Section 7.3.2 lists the supported special character(s) for the composition of administrator passwords, with the following details:

- "The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ">", "[space]", "", "", "+", "-", "/", ":", ";", "<", "=", "?", "[", "\", "]", "[underscore]", "`", "{", "|", "}", and "~")

[ST] Section 7.3.2 states that the minimum_password_length parameter is configurable by a Security Administrator, with the following details:

- "The minimum password length is settable by the Authorized Administrator and can range from 6 to 64 characters."

The listed range includes the value of 15. The listed range includes the value of 15.

Guidance Documentation

The evaluator shall examine the guidance documentation to determine that it:

identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and

provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

[FSAG] Section “Password Policy” outlines the configurable requirements for passwords and provides password best practices.

[FSAG] Section “Configuration limits” lists the min and max values for password creation.

[AGD] Section “Passwords” lists the supported character for password creation.

Tests

The evaluator shall perform the following tests.

a) Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.

b) Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.

Test Number	1
Test Objective	Verify the TOE is able to set a minimum length for the password between 6 and 64 characters and is able to accept any combination of upper and lower case letters, numbers and the special characters selected in [ST].
Test Steps Performed	The evaluator configured the minimum password length on the TOE to be 8 and composed a set of passwords that were at least 8 characters long and together covered all the characters claimed to be supported by the TOE. The TOE accepted each password that met the specified minimum requirements.
Test Result	Pass

Test Number	2
Test Objective	Objective 1: Verify the TOE will not accept password configurations that are less characters than the set ‘minimum password length’ parameter. Objective 2: Verify that the TOE will not accept passwords containing disallowed characters.
Test Steps Performed	The evaluator configured the minimum password length on the TOE to be 8 and composed a password that was less than 8 characters. The TOE rejected the password. The evaluator then configured the password with no special characters, and the TOE rejected it.
Test Result	Pass

2.21 FIA_UIA_EXT.1 User Identification and Authentication

TSS

The evaluator shall examine the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication

mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".

The evaluator shall examine the TSS to determine that it describes which actions are allowed before administration identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not, all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for remote TOE administration and optionally for local TOE administration if claimed by the ST author. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 the TSS shall describe any unauthenticated services/services that are supported by the component.

[ST] Section 7.3.3 describes the logon process for remote authentication mechanism for the local password, SSH password and Web GUI password mechanisms supported by the TOE. The description includes what constitutes a "successful logon" with the following details:

- "The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grants administrative access (if the combination of username and password is correct) or indicates that the login was unsuccessful."

[ST] Section 7.3 describes which actions are allowed before administrator identification and authentication, with the following details:

- "The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed aside from display of the warning banner to the administrator or generate session keys for incoming SSH and TLS connections."
- "The TOE provides for the ability to respond to ICMP echo requests with ICMP echo replies prior to any user identification or authentication."

The TOE is not a distributed TOE.

Guidance Documentation

The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre- shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

[FSGS] Section "GUI" contains instructions to successfully log in the TOE via web interface.

[FSGS] Section "CLI" contains instructions to successfully log in the TOE via console port and SSH.

[AGD] "Enabling administrative access via SSH" contains guidance on logging in via SSH

[FSAG] Sections "Password Policy" and "Administrators" describes configuration of passwords for HTTPS Web GUI and SSH.

Tests

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For all combinations of supported credentials and login methods, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

Test Number	1
Test Objective	Verify that for each of the administrative interfaces, local interface, HTTPS Web GUI, and SSH, the supported credentials and login methods are enforced.
Test Steps Performed	In the evaluated configuration, the TOE supports only remote access with the following credentials: password at Web GUI, SSH CLI and local interface. For each credential and remote access mechanism, the evaluator followed guidance to configure credentials and confirmed that, when providing valid credentials, access to the TOE was granted, and providing invalid credentials resulted in no access to the TOE. The evaluator tested incorrect and correct password credentials on the Web GUI, SSH CLI and local interface.
Test Result	Pass

Test Number	2
Test Objective	a) Verify the login disclaimer is available to the evaluator before authenticating to the HTTPS Web GUI interface b) Verify the login disclaimer is available to the evaluator before authenticating to the SSH administrative interface. c) Verify the HTTPS protocol is successfully established before authenticating to the HTTPS Web GUI interface. d) Verify the SSH protocol is successfully established before authenticating to the SSH administrative interface. e) Verify that only TCP ports 443 (Web GUI) and 22 (SSH) are open on the TOE. f) Verify that no UDP ports are open on the TOE. g) Verify that the TSF responds with ICMP Echo Reply to an ICMP Echo Request.
Test Steps Performed	The evaluator confirmed that the TOE presents a warning banner before authentication. The evaluator confirmed that the HTTPS and SSH protocol were successfully established before authentication, thus performing automatic generation of keys. The evaluator also performed a port scan to verify that the only open ports were the ones used for protocols that required authentication.

	The evaluator verified that the TOE is capable of sending ICMP Echo replies by inspecting traffic captures.
Test Result	Pass

Test Number	3
Test Objective	Determine which services are available to the local administrator before authenticating to the local interface.
Test Steps Performed	The evaluator determined that the login banner was the only service available before authenticating the local interface. The presentation of the login disclaimer is confirmed with FTA_TAB.1 testing.
Test Result	Pass

Test Number	4
Test Objective	The TOE is not distributed
Test Steps Performed	N/A
Test Result	Pass

2.22 FIA_UAU.7 Protected Authentication Feedback

TSS

None.

Guidance Documentation

The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

There are no configuration steps to ensure authentication data is not revealed.

Tests

The evaluator shall perform the following test for each method of local login allowed:

Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

Test Number	1
Test Objective	Verify that the obscured feedback is provided for the local administrator interface.
Test Steps Performed	The evaluator observed that the obscured feedback is provided when inputting the administrator password into the local console.
Test Result	Pass

2.23 FIA_X509_EXT.1/Rev X.509 Certificate Validation (Selection-Based)

TSS

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is

expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected).

The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance.

[ST] Section 7.3.4 describes where the check of validity of the certificates takes place, with the following details:

- "Certificates which are loaded into the trust store for use in an authentication step are evaluated. Since the TOE does not support TLS with mutual authentication, any TLS client certificate presented to the TOE is ignored/not_validated."

[ST] Section 7.3.4 describes when revocation checking is performed and on what certificates, with the following details:

- "All certificates that are presented to the TOE are validated."
- "The TSF performs RFC 5280 certificate validation…"
- "The TOE supports the use of Certificate Revocation Lists (CRLs) as specified in RFC 5280 for revocation status checking of TLS certificates. In all cases, if the revocation status of a certificate cannot be determined, the TSF treats it as invalid. Both the leaf certificate and any intermediate CA certificates are checked for TLS."

[ST] Section 7.3.4 identifies the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE, with the following details:

- "Note that FIA_X509_EXT.1/Rev also optionally requires the TSF to verify the extendedKeyUsage field for certificates used for trusted updates and executable code integrity verification. The TOE does not use X.509 certificates for these purposes so this portion of the requirement is trivially satisfied by the TSF."

The TSS does not state that the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented.

Guidance Documentation

The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describes how certificate revocation checking is performed and on which certificate.

[AGD] Section "X.509 Certificate Operations" describes when the TSF checks certificates for validity; specifically, during TLS communications with remote audit server when the TOE is presented with the server certificate stating that, "Certificates which are loaded into the trust store for use in an authentication step are evaluated. Since the TOE does not support TLS with mutual authentication, any TLS client certificate presented to the TOE is ignored/not_validated".

FIA_X509_EXT.1.1 Tests

The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). The evaluator shall perform the following tests for FIA_X509_EXT.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

Test 3: The evaluator shall test that the TOE can properly handle revoked certificates—conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

Test 4a: [conditional] If OCSP is selected, the evaluator shall configure an authorized responder or use a man-in-the-middle tool to use a delegated OCSP signing authority to respond to the TOE's OCSP request. The resulting positive OCSP response (certStatus: good (0)) shall be signed by an otherwise valid and trusted certificate with the extendedKeyUsage extension that does not contain the OCSPSigning (OID 1.3.6.1.5.5.7.3.9). The evaluator shall verify that the TSF does not successfully complete the revocation check.

Note: Per RFC 6960 Section 4.2.2.2, the OCSP signature authority is delegated when the CA who issued the certificate in question is NOT used to sign OCSP responses.

Test 4b [conditional]: If CRL is selected, the evaluator shall present an otherwise valid CRL signed by a trusted certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC 5280 Section 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

(Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates

(terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate.The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS)

Test Number	1
Test Objective	a) Verify certificate validation succeeds with a valid chain of certificates. b) Verify certificate validation will fail when trust chain is broken.
Test Steps Performed	The evaluator configured the TOE's trust store with a root CA and an intermediate CA. Then presented the TOE with a leaf certificate for validation, which successfully validated. The evaluator then deleted the intermediate CA and CRL from the TOE trust store and again presented a leaf certificate for validation, which was rejected by the TOE.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE rejects an expired certificate.
Test Steps Performed	The evaluator attempted a TLS connection with the TOE using an expired certificate and confirmed that the TOE did not accept this connection.
Test Result	Pass

Test Number	3

Test Objective	1. Verify that when a valid certificate is used, that the validation function succeeds. a. This objective is satisfied by Test 1. 2. Verify the TOE rejects a connection attempt with a peer cert signed by a revoked Intermediate CA. 3. Verify the TOE rejects a connection attempt with a revoked peer cert.
Test Steps Performed	The evaluator attempted TLS connections using both revoked and non-revoked peer and peer intermediate CA certificates to check the revocation status via the CRL. The evaluator confirmed that use of non-revoked certificates resulted in a successful connection and revoked certificates resulted in the TOE denying the connection.
Test Result	Pass

Test Number	4
Test Objective	The TOE doesn't support OCSP. 1. Verify the TOE correctly performs revocation checking when the CRL is signed by CA without CRLsign key usage field.
Test Steps Performed	The evaluator configured the TOE with a CRL that does not contain the crlSign bit and attempted to establish a connection. The TOE failed to establish a connection.
Test Result	Pass

Test Number	5
Test Objective	Verify the TOE rejects a malformed certificate.
Test Steps Performed	The evaluator attempted a TLS connection with a modified byte in the first bytes of the certificate presented to the TOE and verified that the TOE did not accept the connection.
Test Result	Pass

Test Number	6
Test Objective	Verify the TOE rejects a certificate with a corrupted signature.
Test Steps Performed	The evaluator attempted a TLS connection with a modified first byte of the certificate presented to the TOE and verified that the TOE did not accept the connection.
Test Result	Pass

Test Number	7
Test Objective	Verify the TOE rejects a certificate with a modified public key.
Test Steps Performed	The evaluator attempted a TLS connection with a modified byte in the public key of the certificate presented to the TOE and verified that the TOE did not accept the connection.
Test Result	Pass

Test Number	8a
Test Objective	Verify that the TOE is able to establish a successful TLS connection when the leaf certificate along with the EC intermediate CA certificate from outside with the EC Root CA certificate being the only trusted anchor in the TOE.
Test Steps Performed	The evaluator configured a certificate chain consisting of an EC root certificate, an EC intermediate certificate formatted as a named curve, and an EC leaf certificate. The evaluator then imported only the EC root certificate into the TOE. The evaluator configured a test TLS server to present this certificate chain. The evaluator had the TOE attempt to connect to this server and verified that the connection succeeded.

Test Result	Pass

Test Number	8b
Test Objective	Verify that the TOE refuses a TLS connection when presented with an EC Intermediate CA certificate with an explicit curve in the public key information field.
Test Steps Performed	The evaluator configured a certificate chain consisting of an EC root certificate, an EC intermediate certificate formatted as an explicitly-formatted curve, and an EC leaf certificate. The evaluator then imported only the EC root certificate into the TOE. The evaluator configured a test TLS server to present the leaf and intermediate certificates. The evaluator had the TOE attempt to connect to this server and verified that the connection failed.
Test Result	Pass

Test Number	8c
Test Objective	a) Verify the admin can successfully install an Intermediate CA certificate with a named curve into the trust store. b) Verify the TOE rejects an attempt to install an Intermediate CA certificate with an explicit curve, into the trust store.
Test Steps Performed	The evaluator generated two intermediate certificates, one formatted with a named curve, and the second with an explicit format curve. The evaluator was able to load the certificate with the named curve onto the TOE. However, when the evaluator attempted to load the certificate with the explicit format curve, the TOE rejected the certificate.
Test Result	Pass

FIA_X509_EXT.1.2 Tests

Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator shall confirm that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

Test Number	1
Test Objective	Verify the TOE rejects a CA cert without the basicConstraints extension.
Test Steps Performed	The evaluator configured a certificate chain that consisted of a trusted root CA, an intermediate CA that lacked the Basic Constraints extension signed by the root CA, and the leaf certificate signed by the intermediate CA. The

	evaluator then attempted to upload the chain to the TOE and observed the TOE reject the intermediate CA certificate.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE rejects a CA cert with the basicConstraints extension with CA=False when loading to the trust store or verifying the certificate.
Test Steps Performed	The evaluator configured a certificate chain that consisted of a trusted root CA, an intermediate CA that had its CA flag set to FALSE signed by the root CA, and the leaf certificate signed by the intermediate CA. The evaluator configured a test TLS server to present this certificate chain. The evaluator had the TOE attempt to connect to this server and verified that the connection failed.
Test Result	Pass

2.24 FIA_X509_EXT.2 X.509 Certificate Authentication (Selection-Based)

TSS

The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

[ST] Section 7.3.4 describes how the TOE chooses which certificates to use, with the following details:

- "The TOE chooses the certificate to present to external TLS clients based on the server certificate that is loaded into it as part of administrative configuration."

[ST] Section 7.3.4 describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel, with the following details:

- "…if the revocation status of a certificate cannot be determined, the TSF treats it as invalid."

No distinctions between trusted channels are described.

There is no requirement that the administrator be able to specify the default action.

Guidance Documentation

The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

[FSAG] Section "Certificates" describes the steps to manage certificates. The CC mode of operation ensures certificates are used in authentication.

Tests

Test 1: The evaluator shall perform the following test for each trusted channel:

The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

Test Number	1
Test Objective	Verify the TOE performs action described in [ST] ('not accept the certificate') when unable to download a CRL and any cached CRL information is expired.
Test Steps Performed	The test was performed for TLS. The evaluator attempted to establish a TLS connection with a peer whose certificate referenced a CRL distribution point that was going to expire within a few minutes after installing it on the TOE. The evaluator powered down the server hosting the CRL distribution point. The evaluator observed that the TOE attempted to query the CRL distribution point and rejected the connection once the revocation status of the presented certificate could not be established.
Test Result	Pass

2.25 FIA_X509_EXT.3 Extended: X509 Certificate Requests (Selection-Based)

TSS

If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

The ST author did not select "device-specific information.

Guidance Documentation

The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

[AGD] Section "Generating Certificate Requests" contains guidance on generating CSRs. The list of parameters available when generating CSRs is consistent with [ST] Section 6.3.7.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds.

Test Number	1
Test Objective	1) Verify that installation of the appropriate chain of trust into the TOEs trust store enables the TOE to successfully validate a signed CSR upon import into the TOE's trust store. *[This objective is performed here in order to meet Test 2 below]* 2) Verify that the CSR generated by the TOE contains the public key and following fields: a. Common Name b. Organization c. Organizational Unit d. Country
Test Steps Performed	The evaluator generated a certificate request on the TOE and verified it was in the correct format and contained all the information specified by the Security Target.
Test Result	Pass

Test Number	2
Test Objective	1) Verify that certificate validation will fail upon attempt to install a signed CSR into the TOE's trust store when the chain of trust for the signed CSR is not complete. 2) Verify that certificate validation is successful upon attempt to install a signed CSR into the TOE's trust store when the chain of trust for the signed CSR is complete. [This objective is performed in Test 1 above]
Test Steps Performed	The evaluator attempted to import the signed response to the certificate request onto the TOE without the trusted CA imported and confirmed that the import was denied. The evaluator then imported the correct trusted CA and attempted to import the signed response and confirmed that the certificate was successfully imported.
Test Result	Pass

2.26 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

TSS

For distributed TOEs see chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

The TOE is not a distributed TOE.

Guidance Documentation

The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

[FSGS] Section "Update the firmware" contains instructions to manually update the TOE.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.

Test 2: The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

Test Number	1
Test Objective	Verify that authenticated, non-administrative users cannot attempt to update the TOE firmware.
Test Steps Performed	This test is satisfied by the testing performed for "FMT_MOF.1.1/Services – Test 1", as that test confirmed that non-administrative users are unable to perform any administrative actions outside of logging out and changing their own password.
Test Result	Pass

Test Number	2
Test Objective	This test case is covered by FPT_TUD_EXT.1 Test 1.
Test Steps Performed	The test was performed in conjunction to FPT_TUD_EXT.1 Test 1.
Test Result	Pass

2.27 FMT_MOF.1/Services Management of Security Functions Behaviour (Selection-Based)

TSS

For distributed TOEs see chapter 2.4.1.1.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that t operation is performed.

[ST] Section 7.4 lists the services the Security Administrator is able to start and stop and how that operation is performed.

Guidance Documentation

For distributed TOEs see chapter 2.4.1.2.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

[AGD] Section "Services start/stop" lists the services that can be started or stopped by an administrator.

Tests

Test 1: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user

authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Test 2: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.

Test Number	1
Test Objective	Verify that authenticated non-administrative users cannot attempt to disable TSF services.
Test Steps Performed	The evaluator logged into the TOE as a non-administrative user and confirmed that a change to services could not be made without administrative privilege.
Test Result	Pass

Test Number	2
Test Objective	Verify that an authenticated security administrator can enable and disable the Trusted Path TSF (SSH).
Test Steps Performed	The evaluator verified that the administrator could make changes to the TOE's services.
Test Result	Pass

2.28 FMT_MTD.1/CoreData Management of TSF Data

TSS

The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that areaccessiblethroughaninterfacepriorto administratorlog-inareidentified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

There are no administrative functions identified in the guidance documentation that are accessible through an interface prior to administrator log-in. This is supported by the following statement in [ST] Section 7.3:

- "The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed aside from display of the warning banner to the administrator or generate session keys for incoming SSH and TLS connections."

The TOE supports handling of X.509v3 certificates and implements a trust store: The TSS contains sufficient information to describe how the ability to manage the TOE's trust store is restricted, with the following details:

- [ST] Section 7.4 states "The Security Administrator has the ability to manage cryptographic keys to manipulate (add, remove) the certificates (and keys) that reside in the TOE's trust store and to generate certificate signing requests, which include key pairs."

- [ST] Section 7.3 states “The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed aside from display of the warning banner to the administrator or generate session keys for incoming SSH and TLS connections”.
- [ST] Section 7.3.3 states “The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grants administrative access (if the combination of username and password is correct) or indicates that the login was unsuccessful.”

Guidance Documentation

The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

[AGD] Section “Install Updated Certificates” references [FSAG] Section “Certificates” for information regarding certificate management.

[FSAG] Section “Admin Profiles” identifies each of the TSF-data-manipulating functions implemented by the TOE and ensures only administrators have access to them.

[FSGS] Section “Initial Access“ describes that no access to the device is allowed prior to successful identification and authentication.

Tests

No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

2.29 FMT_MTD.1/CryptoKeys Management of TSF Data (Selection-Based)

TSS

For distributed TOEs see chapter 2.4.1.1.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

[ST] Section 7.4 lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and names the operations that are performed, with the following details:

• “The Security Administrator has the ability to manage cryptographic keys to manipulate (add, remove) the certificates (and keys) that reside in the TOE’s trust store and to generate certificate signing requests, which include key pairs.”

Guidance Documentation

For distributed TOEs see chapter 2.4.1.2.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

The TOE is not a distributed TOE.

[FSAG] Section “Certificates” describes the management of signed certificates.

[AGD] Section “Key Zeroization” describes the zeroization of keys and CSPs by factory reset.

[AGD] Section “Signed Certificates” states that “Upon certificate deletion by the administrator, the certificate keys are also deleted”.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Test 2: The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.

Test Number	1
Test Objective	Verify that authenticated non-administrative users cannot manage (generate/import/view/modify/delete) cryptographic keys.
Test Steps Performed	This test is satisfied by the testing performed for “FMT_MOF.1.1/Services–Test 1”, as that test confirmed that non-administrative users are unable to perform any administrative actions outside of logging out and changing their own password.
Test Result	Pass

Test Number	2
Test Objective	Verify that an authenticated security administrator can perform the following actions related to cryptographic keys: a. generate CSR b. import Root CA c. delete Root CA
Test Steps Performed	The evaluator was able to generate a CSR through the WebGUI and was able to import and delete the Root CA successfully.
Test Result	Pass

2.30 FMT_SMF.1 Specification of Management Functions

The security management functions for FMT_SMF.1 are distributed throughout the cPP and are included as part of the requirements in FTA_SSL_EXT.1, FTA_SSL.3, FTA_TAB.1, FMT_MOF.1/ManualUpdate, FMT_MOF.1/AutoUpdate (if included in the ST), FIA_AFL.1, FIA_X509_EXT.2.2 (if included in the ST), FPT_TUD_EXT.1.2 & FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action), FMT_MOF.1/Services, and FMT_MOF.1/Functions (for all of these SFRs that are included in the ST), FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

TSS (containing also requirements on Guidance Documentation and Tests)

The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

(If configure local audit is selected) The evaluator shall examine the TSS and Guidance Documentation to ensure that a description of the logging implementation is described in enough detail to determine how log files are maintained on the TOE.

The evaluator examined the TSS, Guidance Documentation and the TOE as observed during all other testing to confirm that the management functions specified in FMT_SMF.1 are provided by the TOE; the following table lists all management functions claimed in [ST] Section 6.4.5, FMT_SMF.1.1, and identifies which interface the action is available from, the fact that the evaluator verified the functionality was manageable by the security admin and which section within the guidance documentation the details for configuring such functionality resides:

Function	Local	Remote	Evaluator Verified Available During Testing	Guidance Documentation
Ability to administer the TOE locally	X		X	[GSG] Section “To connect to the CLI Console with the GUI:”
Ability to administer the TOE remotely		X	X	[GSG] Section “Connecting to the GUI”
Ability to configure the access banner	X	X	X	[TNO] Section “Admin access disclaimer”
Ability to configure the session inactivity time before session termination	X	X	X	[GSG] Section “Change the GUI idle timeout”

Ability to update the TOE and verify the updates using digital signature capability prior to installing those updates	X	X	X	[GSG] Section “Update the firmware”
Ability to start and stop services		X	X	[AGD] Section “Services start/stop”
Ability to configure authentication failure parameters for FIA_AFL.1	X	X	X	[TNO] Section “Miscellaneous administration related changes”
Ability to manage the cryptographic keys	X	X	X	[ADM] Section “Certificates”
Ability to set the time which is used for time-stamps	X	X	X	[GSG] Section “Configure the system time”
Ability to manage the TOE’s trust store and designate X.509v3 certificates as trust anchors	X	X	X	[ADM] Section “Certificates”
Ability to generate Certificate signing requests (CSRs) and process CA certificate responses	X	X	X	[ADM] Section “Certificates”

[ST] Section 7.4, Table 8, details which security management functions are available through which interface(s).

[ST] Section 7.4 describes the local administrative interface, with the following details:

- “The local interface is a dedicated physical port, direct console interface to the TOE’s CLI. Local access to a virtual TOE is through the ESXi console via URL.”

[GSG] Section “To connect to the CLI through the console port:” describes the local administrative interface.

Since the local console for the pND is truly a local console interface, “warnings for the administrator to ensure the interface is local” are unnecessary.

Guidance Documentation

See section 2.4.4.1

The TOE is not a distributed TOE.

[FSGS] Section “Port and access control information” describes all ports and their interfaces to clearly delineate between interfaces.

[AGD] Section “Functionality Excluded from the Evaluated Configuration” describes the interfaces that should be disabled to remain in the CC evaluated configuration.

[ST] FMT_SMF.1 contains the following management functions:

- Ability to administer the TOE remotely
- Ability to configure the access banner
- Ability to configure the remote session inactivity time before session termination
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates

- o Ability to start and stop services
- o Ability to manage the cryptographic keys
- o Ability to set the time which is used for time-stamps
- o Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors
- o Ability to generate a Certificate Signing Request (CSR) and process the CA certificate response
- o Ability to administer the TOE locally
- o Ability to configure the authentication failure parameters for FIA_AFL.1

[FSAG] Section "Admin Profiles" describes the administrative functions available to each role.

[ST] TSS Section "Security Management" describes the local interface as a dedicated physical port with direct console access to the TOE's CLI. The virtual TOE, when the TOE is a vND, has a local interface described as an "ESXi console via URL".

[AGD] Section "Local Administrative Interface" describes the local console interface for the TOE.

Tests

The evaluator shall test management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

Test Number	1
Test Objective	The following management functions have been tested throughout this test report: • FTA_SSL_EXT.1 – Configuration of the local administrative user idle session timeout period • FTA_SSL.3 - Configuration of the remote administrative users' idle session timeout period • FTA_TAB.1 Configuration of the administrative login notice and concern warning message • FMT_MOF.1/ManualUpdate - Update of the TOE firmware (manual update) • FIA_AFL.1, Configuration password attempt lockout and time period • FPT_TUD_EXT.1.2, Update the TOE manually • FMT_MOF.1/Services – Ability to start and stop services • FMT_MTD.1/CoreData – No separate testing required • FMT_MTD.1.1/CryptoKeys – Management of cryptographic keys • FPT_TST_EXT – Execution of TOE self-tests • FPT_STM_EXT.1 – ability to set time for the time-stamps. • FMT_SMR.2 – ability to administer the TOE locally and remotely • FIA_X509_EXT.1 – ability to manage the TOE trust store • FIA_X509_EXT.3 – ability to generate CSR and process CSR responses.
Test Steps Performed	No additional testing was necessary outside of the SFRs/Tests described in the objectives above.
Test Result	Pass

2.31 FMT_SMR.2 Restrictions on security roles

TSS

The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE (e.g. if local administrators and remote administrators have different privileges or if several types of administrators with different privileges are supported by the TOE).

[ST] Section 7.4 details the TOE supported roles and any restrictions of the roles involving administration of the TOE, with the following details:

- "The TOE supports one default fully privileged user account, Admin, which corresponds to the PP defined Security Administrator. The Admin account has full privileges to all administrative functions."

- "Subordinate Security Administrator accounts may be configured with a variety of permissions and administrative powers, such that the global Admin account may create user accounts to delegate some or all of its powers. In this way, the TOE maintains two roles: the mandatory "Security Administrator" role, and the lesser "TOE User" role. TOE Users may be configured by any security administrator to have some, all, or none of the permissions indicated below."

- Table 8 lists the administrative functions and identifies which administrative interface the function is accessible from.

Guidance Documentation

The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

[FSGS] Section "Initial access" provides configuration for local and remote administration. [AGD] Section "Administration" describes the CC mode of operation as the only configuration needed for remote administration.

Tests

In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; if the TSF shall be validated against the Functional Package for Secure Shell referenced in Section 2.2 of the cPP; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

Test Number	1
Test Objective	Annotate interface usage results during the course of testing.
Test Steps Performed	The evaluator performed this testing throughout overall testing where administrative actions were performed. The testing was generally completed using local interface, SSH console and TLS.
Test Result	Pass

2.32 FPT_APW_EXT.1 Protection of Administrator Passwords

TSS

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

[ST] Section 7.5.1 details all authentication data that are subject to this requirement, with the following details:

- "The TSF stores all Administrative password data in an obfuscated format"

The method used to obscure the plaintext password data when stored is described as follows:

- "The TSF stores all Administrative password data in an obfuscated format. Specifically, the data is stored as a SHA-256 hash."

[ST] Section 7.5.1 describes passwords being stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, with the following details:

- "For both credential and non-public key data, no administrative interface exists to read this data."

Guidance Documentation

None.

Tests

None.

2.33 FPT_SKP_EXT.1 Protection of TSF Data (for Reading of All Pre-shared, Symmetric and Private Keys)

TSS

The evaluator shall examine the TSS to determine that it details how any pre- shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

[ST] Section 7.5.1 details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through any interface designed specifically for that purpose, by any enabled role, with the following details:

- "The TOE stores its SSH private host key in plaintext in flash and does not provide any interfaces to view the key"
- "The TSF stores its private key on its local file system when the key pair for the CSR is first generated. When the signed certificate response is received by the TOE, the certificate and private key are stored in a secure directory that is not accessible to administrators."

For values not stored in plaintext: [ST] Section 7.5.1 describes how they are protected/obscured, with the following details:

- "The TSF stores its private key on its local file system when the key pair for the CSR is first generated (the keypair, generated by the TSF, is created in RAM, then written in plaintext to the non-volatile storage via filesystem APIs). When the signed certificate response is received by the TOE, the certificate and private key are stored in plaintext in a secure directory that is not accessible to administrators."

Guidance Documentation

None.

Tests

None.

2.34 FPT_STM_EXT.1 Reliable Time Stamps

TSS

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

If "obtain time from the underlying virtualization system" is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

[ST] Section 7.5.4 lists each security function that makes use of time, and provides a description of how the time is maintained and considered reliable in the context of each of the time related functions, with the following details:

- "The clock is used for audit record time stamps, measuring session activity for termination, timing of lockout in FIA_AFL.1, and for cryptographic operations based on time/date (e.g. validation of X.509 certificates)"
- "The TOE is a hardware appliance or a virtual appliance image installed on a hardware appliance that includes a hardware-based real-time clock to ensure that reliable time information is available. The TOE's real-time clock stores the system time and date information. The TOE's embedded OS manages the clock and exposes administrator clock-related functions."

The selection of "obtain time from the underlying virtualization system" was not selected.

Guidance Documentation

The evaluator shall examine the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

[FSAG] Section "Configure the system time" contains guidance on how to set the system time.

[AGD] Section "Use of non-CC evaluated features" states that NTP is unevaluated.

Tests

The evaluator shall perform the following tests:

Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator shall use the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator shall observe that the NTP server has set the time to what is expected. If the TOE supports multiple

protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation.

Test 3 [conditional]: If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

Test Number	1
Test Objective	Verify that the system time can be set by the administrator by using the guidance documentation.
Test Steps Performed	The evaluator queried the time on the TOE's WebGUI and then attempted to change the time. The evaluator queried the time again and verified that the time successfully changed.
Test Result	Pass

Test Number	2
Test Objective	Not applicable since [ST] does not claim support for NTP.
Test Steps Performed	N/A
Test Result	Pass

Test Number	3
Test Objective	[ST] did not select "obtain time from the underlying virtualization system", thus this test is not applicable.
Test Steps Performed	N/A
Test Result	Pass

2.35 FPT_TST_EXT.1 TSF Testing

TSS

The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self- tests are run. The evaluator shall also examine the TSS to ensure it describes how the TOE reacts if one or more TOE components fail self-testing (e.g. halting and displaying an error message; failover behaviour).

[ST] Section 6.5.4 identifies the following self-tests:

- verify the integrity of the TOE firmware and software (at power on)

- verify correct operation of cryptographic implementation necessary to fulfil the TSF (prior to providing cryptographic services and on-demand)

- no other self-tests

[ST] Section 7.5.2 details each of the self-tests that are identified by the SFR; this description includes an outline of what the tests are actually doing, with the following details:

- Power on Tests performed (consistent with selections in the SFR):

o "Firmware integrity test"

o "Cryptographic (AES, DHE, SHA, ECDHE, ECDSA, and RSA) known answer tests"

o "SP 800-90A health tests"

o "RNG known answer test"

[ST] Section 7.5.2 makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly, with the following details:

- "These self-tests are sufficient to ensure that the image and configuration file are operating in a known good state, TSF executable code has appropriately not been modified, and that the cryptographic functionality has not been tampered with or degraded, either through a compromise of the cryptographic functionality itself or through a degradation of any hardware components on which this functionality relies."

[ST] Section 6.5.4 states that "The TSF shall respond to all failures". Therefore, [ST] Section 7.5.2 clarifies which response is associated with which type of failure, with the following details:

- "In the event that any of the self-tests or conditional tests fail, the module logs an "error indicator" for the specific test(s) and enters an error state as shown by the following console output"

The TOE is not a distributed TOE.

Guidance Documentation

The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

[AGD] Section "CC Error Mode" describes the event in which one or more self-tests fail and the steps the administrator should take in the event they happen.

[AGD] Section "Entropy" describes the event in which an entropy source failure occurs.

The TOE is not a distributed TOE.

Tests

It is expected that at least the following tests are performed:

Verification of the integrity of the firmware and executable software of the TOE

Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs.

Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to:

[FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.

[FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

The evaluator shall either verify that the self-tests described above are carried out during initial start-up or that the developer has justified any deviation from this.

For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.

Test Number	1
Test Objective	[ST] Section 6.5.4 lists the following self-tests performed by the TSF: a) verification of the integrity of the TOE firmware and software (on startup only) b) verification of the correct operation of the cryptographic implementation necessary to fulfil the TSF (on startup and on demand) The evaluator verifies that objectives a) and b) listed above are performed as described in the [ST] – Note; both objectives are achieved at once when performing the steps. The TOE is not distributed.
Test Steps Performed	The evaluator logged into the local console of the TOE and observed the correct version of the TOE's firmware and software. The evaluator rebooted the TOE via the console and observed that at the end of the boot cycle, the TOE displayed the message, 'Self-tests passed'.
Test Result	Pass

2.36 FPT_TUD_EXT.1 Trusted Update

TSS

The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails.. The evaluator shall verify that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.

If the options 'support automatic checking for updates' or 'support automatic updates' are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator shall examine the guidance documentation instead.

[ST] Section 7.5.3 describes how to query the currently active version, with the following details:

- "The current running version of the firmware can be queried from the CLI using the command: status."

- "An administrator may query the current version of the TOE through the CLI or web interface."

[ST] Section 7.5.3 describes all TSF software update mechanisms for updating the system firmware and software; the description includes a digital signature verification of the software before installation and that installation fails if the verification fails, with the following details:

- "The TOE has a manual update mechanism"

- "Once the update is uploaded to the TOE, a 2048 bit RSA signature is verified for any TOE firmware build (builds include patches) to verify the update is valid"

- "Before proceeding with a TOE update via the GUI or CLI, the following automated process is followed when in the evaluated mode of operation:

o If a signature is not present, abort the upgrade

o Extract the public key and signature from the firmware

o Validate that the public key is the same as is stored on the TOE. If the public keys do not match abort the upgrade.

o Validate the image signature using the public key from the update. If the image validation using the public key fails, abort the upgrade"

[ST] Section 7.5.3 describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification, with the following details:

- "The administrator downloads the TOE to their workstation from https://support.fortinet.com.

- The administrator will then copy the file to the TOE using: HTTPS web interface, SSH, or direct connection to the console port using a RJ-45 to DB-9 serial cable or a nullmodem cable.

- Once the update is uploaded to the TOE, a 2048 bit RSA signature is verified for any TOE firmware build (builds include patches) to verify the update is valid. The signature is compared to a known key value stored on the TOE and pre-configured into the firmware image.

- Successful attempt:

o "After image file is uploaded and its integrity is checked using digital signature, a reboot occurs and the image will be executed."

o "the TOE displays "upgrade successful" and reboots"

- Failed attempts:

- o "If a signature is not present, abort the upgrade"
- o "If the public keys do not match abort the upgrade"
- o "If the image validation using the public key fails, abort the upgrade."

The TOE does not perform a 'delayed activation'.

The TOE does not 'support automatic checking for updates' or 'support automatic updates'.

The TOE is not a distributed TOE.

Guidance Documentation

The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation shall describe how to query the loaded but inactive version.

The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator shall also ensures that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

The TOE does not support delayed activation.

[AGD] Section "Installing the CC firmware build" describes how the administrator can query the active firmware version.

[AGD] Section "Verifying the integrity of the firmware build" describes integrity and authenticity checks at load time by means of an RSA digital signature included in the firmware version.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update

onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator shall verify after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempts to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates:

A modified version (e.g. using a hex editor) of a legitimately signed update

An image that has not been signed

An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)

The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify that both the current version and most recently installed version, reflect the same version information as prior to the update attempt.

The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates).

For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.

Test Number	1
Test Objective	a) Verify that the currently executing TOE firmware version can be queried. *Note: delayed activation is not applicable.* b) Verify that a legitimate update can be successfully installed on the TOE. *Note: No additional activation steps are necessary to install a legitimate update package once the update has been initiated by the admin.* c) Verify that after a successful update, the version verification activity correctly corresponds to that of the update and that current version of the product and most recently installed version match.
Test Steps Performed	The evaluator checked the TOE's version, ran an update and then checked the version again. The TOE was verified to have been updated to the new version.
Test Result	Pass

Test Number	2
Test Objective	a) Verify that the TOE rejects the attempt to install a modified firmware version of a legitimately signed update. b) Verify that the TOE rejects the attempt to install a firmware image that has not been signed. c) Verify that the TOE rejects the attempt to install a firmware image signed with an invalid signature.

Test Steps Performed	The evaluator first confirmed the current version of the TOE. The evaluator modified a byte in the signature section of a candidate update image and attempted to upload it onto the TOE. The TOE rejected the updated image. The evaluator confirmed the TOE version was unchanged. The evaluator then attempted to upload an updated image that had a missing signature. The TOE again rejected the updated image. Lastly, the evaluator then attempted to upload an otherwise valid updated image, with valid signature, that had one byte modified. The TOE rejected this image also. The only update mechanism is manual update, so all testing above was performed using that mechanism. The TOE is not being distributed, so multiple iterations of testing for separate TOE components is not applicable.
Test Result	Pass

2.37 FTA_SSL_EXT.1 TSF-initiated Session Locking

TSS

The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

[ST] Section 7.6 states that local administrative session termination is supported and the related inactivity time period settings, with the following details:

- "A Security Administrator can configure maximum inactivity times for administrative sessions of 1-480 minutes through the TOE GUI and CLI interfaces. A configured inactivity period will be applied to both local and remote sessions in the same manner."
- "When the interface has been idle for more than the configured period of time, the session will be terminated and will require authentication to establish a new session."

Guidance Documentation

The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

[ST] TSS states "A Security Administrator can configure maximum inactivity times for administrative sessions of 1-480 minutes through the TOEs GUI and CLI interfaces".

[AGD] Section "Local Administrative Interface" contains the guidance on configuration of this timeout and this section makes it clear that a single configuration setting for session timeout applies to both the local and remote management interfaces.

[AGD] Section "Local Administrative Interface" states that "When the timeout is reached, the session is terminated".

Tests

The evaluator shall perform the following test:

Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a local interactive session with the TOE. The evaluator shall then observe that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator shall then ensure that re-authentication is needed when trying to unlock the session.

Test Number	1
Test Objective	Verify that the local interactive session with the TOE times out per the inactivity time period referenced in [AGD].
Test Steps Performed	The evaluator set the inactivity period as 2 minutes and 30 minutes and observed that the TOE logged out successfully after 2 and 30 minutes of inactivity (pND) and 2 and 5 minutes (vND).
Test Result	Pass

2.38 FTA_SSL.3 TSF-initiated Termination

TSS

The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

[ST] Section 7.6 details the administrative remote session termination and the related inactivity time period, with the following details:

- "A Security Administrator can configure maximum inactivity times for administrative sessions of 1-480 minutes through the TOE GUI and CLI interfaces. A configured inactivity period will be applied to both local and remote sessions in the same manner."

- "When the interface has been idle for more than the configured period of time, the session will be terminated and will require authentication to establish a new session."

Guidance Documentation

The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

[FSGS] Section "Change the GUI idle timeout" and [FSAG] Section "Settings" outline the steps to configure a session lockout for the HTTPS interface.

[AGD] Section "Remote Administrative Session Timeout" contains instructions on setting the idle timeout values for both SSH and HTTPS Trusted Paths.

Tests

For each method of remote administration, the evaluator shall perform the following test:

Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.

Test Number	1
Test Objective	Verify that the session idle timeout value for each remote administration interface (HTTPS, SSH) can be set and observed.
Test Steps Performed	The evaluator configured the pND TOE to have different inactivity timeout values (2,15 and 30 minutes for the WebGUI and 2 and 20 minutes for the SSH CLI). For vND, the evaluator configured the timeout to 2, 5, and 8 minutes (WebGUI and SSH CLI). The evaluator then authenticated to the TOE and verified that when the inactivity value was hit, the evaluator was

	logged out and had to re-authenticate to the TOE. This was performed for the HTTPS web GUI and SSH CLI.
Test Result	Pass

2.39 FTA_SSL.4 User-initiated Termination

TSS

The evaluator shall examine the TSS to determine that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

[ST] Section 7.6 details how the remote administrative session, and the local administrative session, are terminated, with the following details:

- "Each interface also includes mechanisms for the Security Administrator on each to voluntarily terminate their own session. On the GUI, this is accomplished using a logout button at the top-righthand side where there is UI to sign admin out, or using the 'exit' command on CLI."

Guidance Documentation

The evaluator shall confirm that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

[AGD] Section "Terminating Local and Remote Administration Sessions" describes how the administrator may terminate a local or remote session.

Tests

The evaluator shall perform the following tests:

Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

Test 2: For each method of remote administration, the evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.

Test Number	1
Test Objective	Verify security administrators can terminate their own local session.
Test Steps Performed	The evaluator logged in to the local console and verified that the administrators can terminate their own local session using the 'exit' command.
Test Result	Pass

Test Number	2
Test Objective	Verify that each remote administration method (HTTPS, SSH) can be logged into and terminated by a user.
Test Steps Performed	The evaluator performed a logout on the TOE via the HTTPS web GUI and SSH CLI and confirmed that the user was logged out and needed to re-authenticate to gain access to the TOE.
Test Result	Pass

2.40 FTA_TAB.1 Default TOE Access Banners

TSS

The evaluator shall check the TSS to ensure that it details each administrative method of access (local and/or remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access and might be configured during initial configuration (e.g. via configuration file).

[ST] Section 7.4 details each administrative method of access (local and/or remote) available to the Security Administrator (e.g. serial port, SSH, HTTPS), with the following details:

- Remote:

o "The remote interfaces are an SSH connection to the CLI and a web GUI accessed over TLS/HTTPS."

- Local:

o "The local interface is a dedicated physical port, direct console interface to the TOE's CLI. Local access to a virtual TOE is through the ESXi console via URL."

[ST] Section 7.6 states that the TOE displays an advisory notice and a consent warning message for each administrative method of access, with the following details:

- "Both local and remote interfaces display a configurable pre-authentication warning banner that can be used to advise Security Administrators of appropriate usage of the system."

Guidance Documentation

The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

[AGD] Section "Admin access disclaimer" and [FSAG] Section "Login Disclaimer" provide instructions to enable the login banner.

Tests

The evaluator shall also perform the following test:

Test 1: The evaluator shall follow the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

Test Number	1
Test Objective	Verify that the login banner can be configured by an administrator and is viewed on each administrative interface.
Test Steps Performed	The evaluator configured the TOE access banner and confirmed it was displayed before authentication via the local console, HTTPS web GUI and SSH CLI.
Test Result	Pass

2.41 FTP_ITC.1 Inter-TSF Trusted Channel

TSS

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

[ST] Section 7.7 identifies, for all communications with authorized IT entities listed in the requirement, each secure communication mechanism in terms of the allowed protocols for that IT entity, that the TOE acts as a client, and the method of assured identification of the non-TSF endpoint, with the following details:

- Remote audit server:

o "The TOE supports communications with external audit servers"

o "These connections are protected via a TLS connection where the TOE is a TLS client"

o "TLS provides assured identification of the non-TSF endpoint by validating X.509 certificates."

[ST] Section 7.7 describes all secure communication mechanisms in sufficient detail to allow the evaluator to match it to the FCS_TLSC cryptographic protocol Security Functional Requirements listed in the ST, with the following details:

- "These connections are protected via a TLS connection where the TOE is a TLS client"

Guidance Documentation

The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

[AGD] section "FortiAnalyzer configuration" describes the steps to establish a TLS connection with the log server. [AGD] section "Reconnecting to FortiAnalyzer" describes the steps an administrator needs to take to reconnect in the event of an interruption.

Tests

The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

The evaluator shall perform the following tests:

Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities.

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

The evaluator shall ensure that, when the physical connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

In the case where the TOE is able to detect TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change).

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target.

The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public- facing document or report.

Test Number	1
Test Objective	Verify the TOE is able to establish a trusted channel for the remote audit server.
Test Steps Performed	This test is satisfied by evaluation activities in FCS_TLSC_EXT.1.2 – Test 3.
Test Result	Pass

Test Number	2
Test Objective	Verify the TOE initiates the trusted channel communication between itself and the remote audit server.
Test Steps Performed	This test is covered by evaluation activities in FCS_TLSC_EXT.1.2 – Test 3.
Test Result	Pass

Test Number	3
Test Objective	Verify that when a connection with an authorized IT entity (remote log server) is established, no data is sent in plaintext.
Test Steps Performed	The evaluation team confirmed each such connection was established successfully and that all communicated data was protected.
Test Result	Pass

Test Number	4
Test Objective	Verify that when the network connection between the TOE and a remote server is interrupted that no plaintext data flows, and that, when the network connection is reestablished, data is sent encrypted.

Test Steps Performed	The evaluator had the TOE establish a connection to an external audit server. The evaluator then interrupted the network communication at an intermediate switch level for some time. The evaluator observed that the TOE did not send data in plaintext and maintained protected communications when the connection was restored. The TOE is not distributed.
Test Result	Pass

2.42 FTP_TRP.1/Admin Trusted Path

TSS

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

[ST] Section 7.7 describes the methods of remote TOE administration, along with how those communications are protected, with the following details:

- "Remote CLI sessions occur over an SSH connection."
- "Remote GUI connections take place over a TLS/HTTPS connection."

All protocols listed in [ST] Section 7.7 in support of TOE administration are consistent with those specified in the requirement and are included in the requirements in the [ST] Section 6.2.13 (TLS/HTTPS), and [ST] Sections 6.2.10 and 6.2.11 (SSH).

Guidance Documentation

The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

[AGD] Section "Enabling administrative access via SSH" describes how to enable and configure SSH remote administration.

The TOE's Web GUI HTTPS remote administration is enabled by default. [AGD] Section "Web browser requirements" describes the web browser requirements to establish a connection with the TOE.

Tests

The evaluator shall perform the following tests:

Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.

Test Number	1
Test Objective	a) Verify that an HTTPS connection can be successfully established between the TOE and remote administrative user per the guidance documentation. b) Verify that an SSH connection can be successfully established between the TOE and remote administrative user per the guidance documentation.
Test Steps Performed	This test is satisfied by testing in FCS_HTTPS_EXT, FCS_TLSC_EXT, and FCS_SSH_EXT.1.
Test Result	Pass

Test Number	2
Test Objective	Verify that when a communication channel is established, the traffic is encrypted and no packets are sent in plaintext.
Test Steps Performed	The evaluator initiated a packet capture before starting an HTTPS and SSH session and observed in the packet capture that the traffic is encrypted and no packets are sent in plaintext. The TOE is not distributed.
Test Result	Pass

3 SAR Assurance Activities and Results

3.1 ASE: Security Target Evaluation

3.1.1 General ASE:

When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator ensures the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Result:

For TSS EAs for SFRs, see Section 2 above.

3.1.2 ASE_TSS.1.1C:

The evaluator shall examine the TSS to determine that it is clear which TOE components contribute to each SFR or how the components combine to meet each SFR.

The evaluator shall verify the sufficiency to fulfil the related SFRs. This includes checking that the TOE as a whole fully covers all SFRs and that all functionality that is required to be audited is in fact audited regardless of the component that carries it out.

Result:

The [ST] TSS is consistent with the selections made in the SFRs.

3.2 ADV: Development

3.2.1 Basic Functional Specification (ADV_FSP.1)

The EAs for this assurance component focus on understanding the interfaces (e.g., application programming interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1- 1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional "functional specification" documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

3.2.2 ADV_FSP.1-1

PP Evaluation Activity:

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

Result:

[AGD] Section “Remote access requirements” outlines the TSFI’s as HTTPS, SSH or local console.

[FSGS] Section “CLI” contains guidance on using the local console port.

3.2.3 ADV_FSP.1-2

PP Evaluation Activity:

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

Result:

See ADV_FSP.1-1.

3.2.4 ADV_FSP.1-3

PP Evaluation Activity:

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

Result:
[AGD] and [FSAG] contain guidance on how to use the various functions of the HTTPS GUI interface, SSH console and local console where appropriate. No configuration is necessary to remain compliant other than enabling the CC mode of operation.

3.2.5 ADV_FSP.1-4

PP Evaluation Activity:

Paragraph 609 from the CEM: “In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, this work unit should be considered satisfied.”

Since the rest of the ADV_FSP.1 work units will have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.

Result:
Per [SD] Table 1, since the rest of the ADV_FSP.1 work units will have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.

3.2.6 ADV_FSP.1-5

PP Evaluation Activity:

The evaluator shall examine the interface documentation and confirm that all security-relevant interfaces are described. The evaluator shall also confirm that all internal (FPT_ITT.1), administrative (FTP_TRP.1/Admin), and trusted channel (FTP_ITC.1) requirements unambiguously trace to documented interfaces.

Result:

TSFI to SFR mapping				
SFR	**Name**	**Local console**	**HTTPS GUI**	**SSH**
FAU_GEN.1	Audit Data Generation	X	X	X
FAU_GEN.2	User Identity Association	X	X	X
FAU_STG.1	Protected audit trail storage (Optional)	X	X	X
FAU_STG_EXT.1	Protected Audit Event Storage	X	X	X
FCS_CKM.1	Cryptographic Key Generation (Refinement)	X	X	X
FCS_CKM.2	Cryptographic Key Establishment (Refinement)	X	X	X
FCS_CKM.4	Cryptographic Key Destruction	X	X	X
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	X	X	X
FCS_COP.1/SigGen	Cryptographic Operation (Signature Verification)	X	X	X
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	X	X	X
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	X	X	X
FCS_HTTPS_EXT.1	HTTPS Protocol (Selection-based)		X	
FCS_RBG_EXT.1	Random Bit Generation	X	X	X
FCS_TLSC_EXT.1	TLS Client Protocol (Selection-based)		X	
FCS_TLSC_EXT.2	TLS Client Protocol with Authentication (Optional)		X	
FCS_TLSS_EXT.1	TLS Server Protocol (Selection-based)		X	
FCS_TLSS_EXT.2	TLS Server Protocol with Authentication (Optional)		X	
FCS_SSH_EXT.1	SSH Protocol			X
FCS_SSHS_EXT.1	SSH Protocol - Server			X
FIA_AFL.1	Authentication Failure Management (Refinement)	X	X	X
FIA_PMG_EXT.1	Password Management	X	X	X
FIA_UIA_EXT.1	User Identification and Authentication	X	X	X
FIA_UAU.7	Protected Authentication Feedback	X	X	X
FIA_X509_EXT.1/Rev	X.509 Certificate Validation (Selection-based)	X	X	X
FIA_X509_EXT.2	X.509 Certificate Authentication (Selection-based)	X	X	X
FIA_X509_EXT.3	X.509 Certificate Requests (Selection-based)	X	X	X
FMT_MOF.1/Services	Management of security functions behavior (Selection-based)	X	X	X

TSFI to SFR mapping				
SFR	**Name**	**Local console**	**HTTPS GUI**	**SSH**
FMT_MOF.1/ManualUpdate	Management of security functions behavior	X	X	X
FMT_MTD.1/CoreData	Management of TSF Data	X	X	X
FMT_MTD.1/CryptoKeys	Management of TSF data (Selection-based)	X	X	X
FMT_SMF.1	Specification of Management Functions	X	X	X
FMT_SMR.2	Restrictions on security roles	X	X	X
FPT_APW_EXT.1	Protection of Administrator Passwords	X	X	X
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)	X	X	X
FPT_STM_EXT.1	Reliable Time Stamps	X	X	X
FPT_TST_EXT.1	TSF Testing (Extended)	X	X	X
FPT_TST_EXT.3	TSF Self-Test with Defined Methods	X	X	X
FPT_TUD_EXT.1	Trusted Update	X	X	X
FTA_SSL_EXT.1	TSF-initiated Session Locking		X	X
FTA_SSL.3	TSF-initiated Termination (Refinement)	X	X	X
FTA_SSL.4	User-initiated Termination (Refinement)	X	X	X
FTA_TAB.1	Default TOE Access Banners (Refinement)	X	X	X
FTP_ITC.1	Inter-TSF trusted channel (Refinement)	X	X	X
FTP_TRP.1/Admin	Trusted Path (Refinement)	X	X	X

3.2.7 ADV_FSP.1-6

PP Evaluation Activity:

EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e. at the TSFI) are covered. Therefore, the intent of this work unit is covered.

Result:
Per [SD], these activities are covered by other evaluation activities.

3.2.8 ADV_FSP.1-7

PP Evaluation Activity:

EAs that are associated with the SFRs in Section 2, and, if applicable, Sections 3 and 4, are performed to ensure that all the SFRs where the security functionality is externally visible (i.e. at the TSFI) are addressed, and that the description of the interfaces is accurate with respect to the specification captured in the SFRs. Therefore, the intent of this work unit is covered.

Result:
Per [SD], these activities are covered by other evaluation activities.

3.3 AGD: Guidance Documents

3.3.1 Operational User Guidance (AGD_OPE.1)

The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR. For the related evaluation activities, the evaluation evidence documents Security Target, AGD documentation (user guidance) and functional specification documentation (if provided) shall be used as input documents. Each input document is subject to ALC_CMS.1-2 requirements.602. In addition, the evaluator performs the EAs specified below.

PP Evaluation Activities:

The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Result:

[AGD] Section “Introduction” refers to the Fortinet FortiSandbox Documentation Set which includes the [AGD], [FSAG] and [FSGS].

[CFG] configuration list (proprietary) provided by the vendor contains the complete list of documents that constitute the CC user guidance documentation. This list is reflected in [ST] Section 9, Table 8 which contain the same references to documentation that constitute the guidance documentation for the TOE.

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Result:

[ST] claims two (2) operational environments, the physical network device and the virtual network device (vND) environment.

[AGD], [FSGS], [FSAG] provide specific and general operational guidance for both the physical and virtual environments.

[IGV] provides specific guidance regarding the virtual network device.

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

Result:

The [AGD] only describes the use of the CC mode of operation, which enforces [ST] selected cryptographic implementations.

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Result:

[AGD] describes three (3) management interfaces, the local console, HTTPS Web GUI and the optional SSH interface.

[AGD] claims only the CC mode of operation as the evaluated configuration and all other security functionalities are unevaluated.

In addition the evaluator shall ensure that the following requirements are also met.

The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The evaluator shall verify that this process includes instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Result:

[AGD] Section “Enabling CC mode” states that enabling the CC mode of operation will enable strict defaults. The cryptographic implementations are not configurable.

[AGD] Section “Functionality Excluded from the Evaluated Configuration” describes the functionality that will be disabled and if enabled will cause the TOE to be in an unevaluated state.

[AGD] Section “Downloading the CC certified firmware” describes the steps required to download the firmware image.

[AGD] Section "Installing the CC firmware build” describes installing the firmware to the TOE.

[AGD] outlines all modified security functionality that is available in the TOE once the CC mode of operation is enabled. Once the CC mode of operation is enabled [ST] selections are enforced.

[AGD] Section “Use of non-CC evaluated features” identifies unevaluated functionality. This section also instructs the admin to review [ST] to ensure they understand which functionality was evaluated.

[AGD] Section “Functionality covered by CC evaluation activities” identifies security functionality that has been evaluated.

3.3.2 Preparative Procedures (AGD_PRE.1)

PP Evaluation Activities:

The evaluator shall examine the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

Result:

ST] defines the following security objectives:

- OE.PHYSICAL, OE.TRUSTED_ADMIN

- [AGD] Section “Installation requirements” provides installation requirements to the admin, instructing them to install the TOE in a secure physical location, and such installation is restricted to authorized operators.
- [IGV] Section “Deployment” contains guidance on configuring the vND’s hypervisor to accept the virtual TOE.
- [AGD] Sections “Potential Firmware issues” and “Potential Hardware issues” outline potential problems with installation and to contact Fortinet technical support.
- [FSGS] Section “Checklist” and its subsections contain the steps necessary to perform the initial configuration of the TOE, including changing the default password and settings.
- [AGD] Section “The CC Mode of Operation” contains guidance on enabling the CC mode of operation to force conformance to [ST] selections.

- OE.VM_CONFIGURATION
  - [IGV] Sections “Preparing for deployment” and “Deployment” contain guidance on correctly implementing the VM
  - [AGD] Section “VM Security” addresses guidance procedures pertaining to maintaining security.
- OE.RESIDUAL_INFORMATION
  - [AGD] provides information on key zeroization during factory reset.
- OE.NO_GENERAL_PURPOSE
  - Not applicable for preparation phase.
- OE.UPDATES
  - Not applicable for preparation phase.
- OE.ADMIN_CREDENTIALS_SECURE
  - Not applicable for preparation phase.
  - The Evaluator concluded that Guidance documentation is written with sufficient details to be used and understood by the target audience.

The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Result:

[ST] Section 1.4 describes the TOE’s hardware, software, and OE requirements.

[AGD] Section “Installation Requirements” describes the operational environment requirements for the pND.

[IGV] Sections “Preparing for deployment” and “Deployment” cover operational environment requirements for the vND.

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

Result:

[AGD] Section “Installing the unit” references the “Fortinet QuickStart Guide” and the “Getting Started” section of the Fortinet Technical Docs website for physical installation of the TOE. The [ST] claims two Operational Environments, the physical and vND.

[IGV] Sections “Preparation for Deployment” and “Deployment” cover the configuration and deployment of the virtual machine TOE image.

The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.

The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e.,evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1,

FMT_MTD.1 and FMT_MOF.1 series of SFRs.

Result:

[AGD] Section “Introduction” explains the TOE’s role in the larger customer infrastructure.

[FSGS] Sections “Initial Access” and “Checklist” outline the first steps in configuration of the TOE.

[AGD] Section “The CC Mode of Operation” describes placing the TOE in the CC mode of operation.

[AGD] Section “Log specific settings” describes the configuration for remote log servers.

[IGV] Section “Preparation for Deployment” and “Deployment” contain preparative procedures for the vND operational environment.

In addition the evaluator shall ensure that the following requirements are also met.

The preparative procedures must:

a) include instructions to provide a protected administrative capability; and

b) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

Result:

[AGD] Section “Remote access requirements” describes the secure administration interfaces of the TOE, using local management, HTTPS or SSH.

[FSGS] Section “Change the administrator password” states the default values associated with the default administrator account and includes instructions on how to change it.

3.4 ALC: Life-cycle Support

3.4.1 Labelling of the TOE (ALC_CMC.1)

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

Result:

The evaluator examined the TOE and verified that it has labeling containing a model identification (see [ATE] Section 7, Figures 1-16).

3.4.2 TOE CM Coverage (ALC_CMS.1)

3.4.3 ALC_CMS.1-1

The evaluator shall check that the configuration list includes the following set of items:

a) the TOE itself;

b) the evaluation evidence required by the SARs in the ST.

[CFG] contains a configuration list that includes TOE model identification and firmware version.

[CFG] contains a list of evidence provided for evaluation: Security Target, Guidance, and Entropy Analysis Report.

3.4.4 ALC_CMS.1-2

The evaluator shall examine the configuration list to determine that it uniquely identifies each configuration item.

The configuration list contains sufficient information to uniquely identify which version of each item has been used (typically a version number). Use of this list will enable the evaluator to check that the correct configuration items, and the correct version of each item, have been used during the evaluation.

[CFG] contains identification for TOE hardware versions and firmware version.

[CFG] contains the date and version for the Security Target, Guidance documents, and Entropy Analysis Report.

3.5 ATE: Tests

3.5.1 Independent Testing (ATE_IND.1)

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator should consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation. Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section B.4.3.1.

Result:

- The administrative guidance documentation was utilized to configure the TOE into its evaluated configuration. The evaluator ensured that the actual guidance documentation used matched the description in the [ST].
- The physical TOE utilized matched the description in [ST] Section 1.4 and contains the description of the physical scope of the TOE. The evaluator examined the TOE and determined that it is installed properly and is in a known good state.
- The evaluator used [SD] and [PKG] to devise an appropriate Test Plan.
- The evaluator performed testing according to the test plan. This document contains evidence of the testing performed.
- The evaluator designed a test plan that is reproducible and describes the necessary instructions to perform the tests, communicate with the TOE, and observe the results. It contains expected test results and actual test results.
- The evaluator observed that actual test results are consistent with the expected test results.

- The testing approach, configuration, depth, and results of testing have been incorporated into ETR.

3.6 AVA: Vulnerability Assessment

3.6.1 Vulnerability Survey (AVA_VAN.1)

While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator shall follow a set of well-defined activities and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

Evaluation Activities:

The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator shall document the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Result:

For Type 1 vulnerabilities, the Evaluator conducted a search using the PP-prescribed sources (Section A.4) for relevant keywords pertaining to the TOE (keywords prescribed by the PP and those formulated by the required vendor provided documentation). Search results were screened and hits that bear no relation to the evaluated technology (specifically where module or product names were shared but related to another class of products) were removed from consideration – the remaining results were retained for further analysis.

For Type 2 vulnerabilities, Additional Flaw Hypotheses test was performed for Bleichenbacher style attacks against TLS, with no vulnerabilities found.

No Type 3 vulnerability flaw hypotheses were generated; thus, no Type 3 vulnerabilities were identified.

Type 4 vulnerability testing (fuzzing) was not performed.

No vulnerabilities were identified using the search terms as required by "NIAP Policy 17 Addendum – Vulnerability Mitigation Guidance" documentation.

Based on the vulnerability analysis performed, the evaluation team determined that no residual vulnerabilities exist in the product that are exploitable by attackers with Basic Attack Potential.

4 Cryptographic Certificate Mapping

SFR	Algorithm	CAVP #	NIST Standard
FCS_CKM.1	RSA	A7080, A7082	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3
FCS_CKM.1	ECDSA	A7080, A7082	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4
FCS_CKM.1	FFC	A7080, A7082	NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”
FCS_CKM.2	ECC	A7080, A7082	NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”
FCS_CKM.2	FFC	A7080, A7082	NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”
FCS_COP.1/DataEncryption	AES-CBC	A7080, A7081, A7082, A7083	NIST SP 800-38A
FCS_COP.1/DataEncryption	AES-GCM	A7080, A7081, A7082, A7083	NIST SP 800-38D
FCS_COP.1/SigGen	RSA	A7080, A7082	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4
FCS_COP.1/SigGen	ECDSA	A7080, A7082	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5
FCS_COP.1/Hash	SHA-1	A7080, A7081, A7082, A7083	FIPS Pub 180-4
FCS_COP.1/Hash	SHA-2	A7080, A7081, A7082, A7083	FIPS Pub 180-4
FCS_COP.1/KeyedHash	HMAC	A7080, A7081, A7082, A7083	FIPS Pub 198-1, "The KeyedHash Message Authentication Code, and FIPS Pub 180-4, “Secure Hash Standard or ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2
FCS_RBG_EXT.1	Counter-DRBG	A7081, A7083	NIST SP 800-90A

5 References

Abbr.	Name	Version	Date
[PP]	collaborative Protection Profile for Network Devices	3.0e	June 4, 2023
[SD]	Supporting Document Mandatory Technical Document. Evaluation Activities for Network Device cPP	3.0e	June 4, 2023
[PKG]	Functional Package for Secure Shell (SSH)	1.0	2021-05-13
[ST]	FortiSandbox 4.4 Security Target	1.6	October, 2025
[AGD]	FortiSandbox 4.4 NDcPP Common Criteria Technote	34-446-1080542-20251027	October 30, 2025
[FSAG]	FortiSandbox 4.4.6 Administration Guide	34-445-1030730-20240523	December 20, 2024
[FSGS]	FortiSandbox 4.4.6 Getting Started	34-444-1011093-20250731	December 20, 2024
[IGV]	FortiSandbox 4.4.0-4.4.7Install Guide for VMware	34-445-888478-20250221	February 21, 2025
[LOG1]	FortiSandbox 4.4.0 Log Reference	34-440-922625-20230711	July 11, 2023
[LOG2]	NDcPP Common Criteria Logging Addendum FortiSandbox 4.4.6	N/A	2025